



TAMPEREEN TEKNILLINEN YLIOPISTO

MEERI PANULA

PARHAAT KÄYTÄNNÖT STACKIN KÄYTTÖÖN AUTOMAATTI-  
SESTI ARVIOITAVIEN MATEMATIIKAN TEHTÄVIEN LUOMISEEN  
Diplomityö

Tarkastajat: dosentti Ossi Nykänen  
lehtori Janne Kauhanen  
Tarkastaja ja aihe hyväksytty  
Tieto- ja sähkötekniikan tiedekunta-  
neuvoston kokouksessa 9. touko-  
kuuta 2012

## TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

**PANULA, MEERI:** Parhaat käytännöt STACKin käyttöön automaattisesti arvioidavien matematiikan tehtävien luomiseen

Diplomityö, 69 sivua

Toukokuu 2012

Pääaine: Hypermedia

Tarkastajat: dosentti Ossi Nykänen, lehtori Janne Kauhanen

Avainsanat: STACK, matematiikan tehtävät, automaattinen arviointi, käyttöohjeet, MathBridge, verkko-opetus

Matematiikan verkko-opetukseen voidaan tuoda lisäarvoa tarjoamalla opiskelijoille automaattisesti tarkastettavia matematiikan tehtäviä. Tässä työssä tarkastellaan tällaisten tehtävien luomista STACK-järjestelmällä. Työn tavoitteet ovat selvittää, millainen järjestelmä STACK on, kuinka STACKia tulisi käyttää teknisesti ja opetusjärjestelyjen näkökulmasta ja millaisia ohjeita STACKia käyttävälle opettajalle pitäisi antaa.

Aluksi esitellään työn konteksti kertomalla matematiikan verkko-opetuksesta yleisesti ja Tampereen teknillisellä yliopistolla. Seuraavaksi esitellään STACKin eri versiot ja tarkastellaan lähemmin STACK2:n ominaisuuksia ja teknistä toteutusta sekä arvioidaan sen soveltuvuutta opetuskäyttöön. STACKia testattiin osana Insinöörimatematiikka B 4u -opintojakson kevään 2012 toteutusta. Tulosten perusteella tehdään ylläpitosuunnitelma ja annetaan jatkokehitysideoita. Lopuksi esitellään työn päätuloksena syntyneet opettajan ohjeet.

Tutkimus osoittaa, että STACKilla on mahdollista luoda yliopistotasoisia, opetuksen kannalta mielekkäitä matematiikan tehtäviä. STACK ei kuitenkaan kerro opettajalle, mitä tai miten tulisi opettaa, eikä pelkkä järjestelmän olemassaolo motivoi opiskelijoita käyttämään sitä. STACKin käyttöä hankaloittavat epäintuitiiviset käyttöliittymät, useat käytettävät syntaksit ja ajonaikaiset tekniset ongelmat. Opettajan ohjeet esittelevät STACKin ominaisuuksia ja antavat konkreettisia ohjeita sen käyttöön.

## ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

**PANULA, MEERI:** Best Practices in Using STACK for Creating Automatically Assessed Mathematics Exercises

Master of Science Thesis, 69 pages

May 2012

Major: Hypermedia

Examiners: Adjunct Professor Ossi Nykänen, Lecturer Janne Kauhanen

Keywords: STACK, mathematics exercises, automatic assessment, user manual, MathBridge, e-learning

Automatically assessed exercises can be used to enhance online learning of mathematics. This thesis examines creating exercises with the open-source system STACK. The goals are to understand what kind of system STACK is, how STACK should be used technically and as a learning environment and what kind of advice should be given to teachers using STACK.

First the context of the thesis is explained by studying online learning of mathematics in general and at Tampere University of Technology. All the versions of STACK are introduced and a closer look is taken at STACK2's features and technologies. STACK2 is also assessed. The results of tests using STACK at Insinöörimatematiikka B 4u course are introduced and used to create plans for maintenance and development. The main result of the study, a collection of best practices for teachers, is explained.

The study shows that STACK can be used to create university level, useful mathematics exercises. However, STACK cannot tell teachers what or how to teach, and the existence of the system alone is not enough to motivate students to use it. The usage of STACK is hindered by unintuitive user interfaces, the number of syntaxes needed to create exercises and run-time technical problems. The collection of best practices for teachers introduce STACK's features and give concrete advice to users.

## ALKUSANAT

Tämä diplomityö tehtiin Tampereen teknillisen yliopiston Matematiikan laitokselle. Haluan kiittää lämpimästi Ossi Nykästä ja Janne Kauhasta työn aikana saadusta ohjauksesta ja kommentteista. Kiitän myös Teemu Mäkelää avusta MathBridgen kanssa ja sitä käsittelevän luvun kommentoinnista, Matti Harjulaa avusta OtaSTACKin kanssa ja nykyisiä ja entisiä työtovereita Matematiikan laitoksella ja Tieto- ja sähkötekniikan tiedekunnassa.

Lopuksi haluan kiittää tuesta ja kannustuksesta perhettäni ja ystäviäni.

Tampereella 21. toukokuuta 2012

Meeri Panula

# SISÄLLYS

1	Johdanto .....	1
2	Tausta .....	3
	2.1 Verkko opetuksessa.....	3
	2.2 Matematiikan tehtävät verkko-opetuksessa .....	5
	2.3 Matematiikan verkko-opetusjärjestelmien arviointi .....	9
	2.4 Matematiikan verkko-opetus Tampereen teknillisellä yliopistolla .....	13
	2.5 MathBridge .....	15
	2.6 Vaatimukset verkko-opetusjärjestelmälle .....	18
3	STACK.....	21
	3.1 Historia ja versiot .....	21
	3.1.1 STACK1 .....	21
	3.1.2 OtaSTACK.....	22
	3.1.3 STACK2 .....	23
	3.1.4 STACK3 .....	23
	3.1.5 Vertailu .....	24
	3.2 Tekninen kuvaus .....	27
	3.2.1 Opiskelijan käyttöliittymä.....	28
	3.2.2 Tehtävän luominen .....	30
	3.2.3 Tehtävän tallennus .....	37
	3.2.4 Matematiikkaohjelmisto .....	40
	3.2.5 Moodle-integraatio.....	40
	3.3 Arviointia .....	41
4	STACK käytössä.....	45
	4.1 Tehdyt muutokset.....	45
	4.2 Testaus.....	45
	4.2.1 Testi 1 .....	46
	4.2.2 Testi 2 .....	48
	4.3 Ylläpitosuunnitelma .....	50
5	Opettajan ohjeet .....	53
	5.1 Opettajan rooli.....	53
	5.2 Ohjeiden tavoitteet .....	54
	5.3 Ohjeiden toteutus .....	55
	5.4 Ohjeiden sisältö.....	55
	5.4.1 Ylläpitäjän työkalut.....	56
	5.4.2 Moodle-opintojakson hallinta.....	57
	5.4.3 STACK-tehtävän suunnittelu.....	58
	5.4.4 STACK-tehtävän luominen .....	59
	5.4.5 STACK-tehtävien lisääminen Moodleen.....	61
	5.4.6 Opiskelijoiden vastausten tarkastelu.....	61
6	Johtopäätökset.....	63

Lähteet.....66

## TERMIT JA NIIDEN MÄÄRITELMÄT

ActiveMath	Eräs matematiikan verkko-oppimisalusta.
HTML, XHTML	(eXtensible) HyperText Mark-up Language. Verkkosivun rakenteen määrittäviä merkintäkieliä.
Insinöörimatematiikka 1-4	Tampereen teknillisellä yliopistolla luennoitava opintojakso, joka on pakollinen suurimmalle osalle opiskelijoista. Suositellaan suoritettavaksi ensimmäisenä opintovuonna.
JavaScript	Komentosarjakieli, jota käytetään lisäämään verkkosivuille dynaamista toiminnallisuutta.
jsMath	Eräs matemaattisten merkintöjen esittämiseen tarkoitettu JavaScript-kirjasto.
LaTeX	Eräs ladontajärjestelmä, jota käytetään erityisesti matemaattisia merkintöjä sisältävän tekstin esittämiseen.
matematiikkajumppa	Tampereen teknillisellä yliopistolla järjestettävä verkko-opintojakso, jolla kerrataan lukiomatematiikkaa.
matematiikkaohjelmisto, CAS	Tietokoneohjelma, jolla voidaan käsitellä matemaattisia lausekkeita ja suorittaa laskutoimituksia.
MathBridge	(1) EU-projekti, jonka tavoite oli parantaa uusien yliopisto-opiskelijoiden matemaattisia taitoja. (2) Edellisessä toteutettu matematiikan verkko-oppimisalusta.
MathJax	Eräs matemaattisten merkintöjen esittämiseen tarkoitettu JavaScript-kirjasto.
MathML	Mathematical Markup Language. Matemaattisten merkintöjen esittämistä varten suunniteltu merkintäkieli.
Maxima	Eräs avoimen lähdekoodin matematiikkaohjelmisto.
Moodle	Eräs avoimen lähdekoodin oppimisympäristö.
MySQL	Eräs avoimen lähdekoodin tietokannanhallintajärjestelmä.
perustaitojen testi	Uusien opiskelijoiden Tampereen teknillisellä yliopistolla suorittama tietokoneavusteinen matematiikan lähtötasotesti.
plugin	Tekniikka, joka mahdollistaa sovelluksen käyttämisen toisen sovelluksen kautta.
SOAP	Simple Object Access Protocol. Sovellusten väliseen datavaihtoon käytetty tekniikka.
STACK	System for Teaching and Assessment using a Computer algebra Kernel. Eräs järjestelmä interaktiivisten, automaattisesti tarkastettavien matematiikan tehtävien luomiseen.
tarkastuspuu	Rakenne, jota STACK käyttää tehtävien tarkastamiseen.
TTY	Tampereen teknillinen yliopisto.
wiki	Verkkosivusto, jonka sisältö on käyttäjien muokattavissa.
XML	eXtensible Markup Language. Merkintäkieli, jolla kuvataan dokumentin rakenne.

XML-skeema

XML-dokumentin rakenteen kuvaava teknologia.



# 1 JOHDANTO

Internet on 2000-luvulla noussut merkittävään osaan suomalaisten yliopistojen opetuksessa. Saataville on tullut suuri valikoima sekä kaupallisia että ilmaisia verkko-opetuksen työkaluja. Kaikkia opetettavia asioita ei kuitenkaan toistaiseksi voida käsitellä yhden yleisen verkko-opetusjärjestelmän avulla. Matemaattisten aineiden verkko-opetuksen tavallisimmat haasteet liittyvät matemaattisten merkintöjen esittämiseen, matemaattisen syntaksin määrittelyyn ja vastausten oikeellisuuden arvioimiseen. Näitä ongelmia ratkomaan on kehitetty erilaisia juuri matematiikanopetukseen tarkoitettuja verkko-opetusjärjestelmiä, joista osa toimii itsenäisinä sovelluksina, osa integroituu muihin oppimisalustoihin.

Tampereen teknillisen yliopiston matematiikanopetuksessa verkko-opetuksella ei toistaiseksi ole ollut suurta roolia. Kiinnostus kyseisen järjestelmän käytön laajentamiseen tai uuden järjestelmän käyttöönottoon on kuitenkin herännyt. Verkko-opetuksen toivotaan parantavan oppimistuloksia erityisesti matematiikan peruskursseilla, joilla keskeytysprosentti on suuri.

Tämä diplomityö tarkastelee ilmaista, avoimen lähdekoodin System for Teaching and Assessment using a Computer algebra Kernel -järjestelmää (STACK), jota käytetään interaktiivisten, automaattisesti tarkastettavien matematiikan tehtävien luomiseen. Aihetta tarkastellaan erityisesti tehtävät toteuttavan matematiikanopettajan näkökulmasta. Tutkimuksen tavoitteena on selvittää:

- millainen järjestelmä STACK on
- kuinka STACKia tulisi käyttää teknisesti
- kuinka STACKia tulisi käyttää käytännön opetusjärjestelyjen näkökulmasta ja
- millaisia ohjeita STACKia käyttäville opettajille pitäisi antaa.

Näiden kysymysten vastauksista on hyötyä sekä STACKin nykyisille käyttäjille että STACKin käyttöönottoa harkitseville oppilaitoksille.

Tutkimuksen aluksi luodaan konteksti tutustumalla matematiikan verkko-opetusjärjestelmien yleisiin piirteisiin ja merkittäviin sovelluksiin. Sen jälkeen tutustutaan TTY:llä järjestettävään matematiikan verkko-opetukseen ja siinä käytettävään MathBridge-järjestelmään. Sen jälkeen STACKin eri versiot esitellään ja niiden soveltuvuutta opetuskäyttöön TTY:llä arvioidaan. Tämän vertailun perusteella valitaan yksi versio, jonka toteutustekniikoita ja käyttöä tarkastellaan lähemmin.

Matematiikan verkko-opetusjärjestelmiä varten on kirjoitettu joitakin käyttöoppaita. Tehtävien luomisen pedagogiikkaa ei juurikaan käsitellä. Tutkimuksessa selvitetään, kuinka STACKin tekniset ominaisuudet mahdollistavat erilaisia opetuksellisia valintoja ja millaisia valintoja opettajan tulee tehdä hyödyntääkseen STACKin tarjoamia mahdollisuuksia.

Tutkimuksen päätulos on STACKia käyttäville matematiikanopettajille suunnattu ohjekokoelma, jonka sisältö esitellään ja perustellaan. STACKin soveltuvuutta opetuskäyttöön Tampereen teknillisellä yliopistolla arvioidaan. Tutkimukseen sisältyy myös hieinan ohjelmistokehitystä STACKin mukauttamisen muodossa sekä STACKin testausta matematiikan opintojaksolla.

## 2 TAUSTA

Jotta tietokoneohjelman toimintaa voidaan ymmärtää ja arvioida, on ensin tutustuttava ongelmiin, joita sen käytöllä halutaan ratkoa, ja ympäristöön, jossa sitä käytetään. Tämä luku kertoo verkko-opetuksesta ja matematiikan erityispiirteistä, esittelee erilaisia matematiikan verkko-opetusjärjestelmiä ja tekee katsauksen matematiikan verkko-opetuksen nykytilaan ja tulevaisuuden toiveisiin Tampereen teknillisellä yliopistolla.

### 2.1 Verkko opetuksessa

Termi verkko-opetus sisältää kaiken opetuksen, joka hyödyntää Internetin tarjoamia palveluita, kuten WWW:tä, sähköpostia tai verkossa toimivia ohjelmia (Pantzar 2004). Internetiä voidaan hyödyntää opetuksessa eri laajuuksissa. Taulukko 2.1 kuvaa verkon roolia erityyppisillä opintojaksoilla. Verkko voi olla lähi- tai etäopetusta tukeva työkalu, yksi monimuoto-opetuksen tasa-arvoisista työkaluista, sulautuvan opetuksen välttämätön työkalu tai aidossa verkko-opetuksessa ainoa työkalu. Tässä työssä termiä verkko-opetus käytetään kuvaamaan verkossa tapahtuvaa opetusta riippumatta siitä, liittyykö opintojaksoon lisäksi muita opetusmuotoja, kuten lähiopetusta.

Verkko-opetus voi tapahtua synkronisesti tai asynkronisesti. Synkronisessa kommunikaatiossa opiskelija ja opettaja kommunikoivat reaaliaikaisesti, esimerkiksi chat-huoneessa joka maanantai klo 14–15. Asynkronisessa kommunikaatiossa osapuolet työskentelevät toisistaan riippumattomina aikoina: Opettaja lisää opintojaksolla käsiteltävän materiaalin verkkoon lukukauden alkaessa, ja opiskelija tutustuu siihen viikko ennen tenttiä.

Synkronisten opetusmuotojen suurin etu on niiden paikkariippumattomuus. Opettaja ja opiskelijat voivat vaihtaa ajatuksia videoneuvottelussa, vaikka kaikki asuisivat eri mantereilla. Asynkronisuus tekee opetuksesta paikkariippumattoman lisäksi myös aikariippumatonta. Tietotekniikan rooli ihmisten arjessa on hiipinyt myös opetukseen: Internetin ansiosta tiedonhaku ei rajoitu enää kirjaston aukioloaikoihin ja sähköposti kulkee pyhäpäivinäkin. Moni haluaa opiskella, vaikka työssäkäynti, lastenhoito tai aktiivinen harrastus estää luentoaikoihin sitoutumisen. Verkko-opetus tarjoaa opiskelijalle mahdollisuuden tutustua materiaaliin ja tehdä harjoituksia hänelle itselleen parhaiten sopivaan aikaan, mikä voi tehdä verkko-opetusta tarjoavista yliopistoista hakijoille houkuttelevampia ja lyhentää opiskeluaikoja. Synkroninen opetus soveltuu erityisesti kohtalaisen yksinkertaisista asioista keskusteluun, tutustumiseen ja tehtävien suunnitteluun, asyn-

kroninen opetus monimutkaisten asioiden pohtimiseen ja aikataulukonfliktien välttämiseen (Hrastinski 2008).

**Taulukko 2.1.** Verkon rooli opetuksessa erityyppisillä opintojaksoilla.

<b>Verkon rooli opetuksessa</b>	<b>Esimerkki</b>	<b>Opintojakson tyyppi</b>
Opetusta tukeva	Opintojakson kotisivulla on nähtävillä luentoaikataulu. Opiskelija voi esittää opettajalle kysymyksiä sähköpostilla.	Lähiopetus Etäopetus (esimerkiksi radion, television tai postin välityksellä)
Yksi opetuksen työkaluista	Opiskelija voi suorittaa pakollisen harjoituksen osallistumalla yliopistolla järjestettävälle harjoitustunnille tai tuottamalla sisältöä opintojakson wikiin.	Monimuoto-opetus
Opetuksen välttämätön työkalu	Opiskelijan oppiminen ei rajoitu erillisiin opetus- ja opiskelutilanteisiin, vaan sitä tapahtuu jatkuvasti opiskelijan käyttäessä Internetiä osana jokapäiväistä elämäänsä.	Sulautuva opetus (Itkonen-Isakov 2009, Allan 2007)
Opetuksen ainoa työkalu	Ulkomailla oleskeleva opiskelija suorittaa koko opintojakson verkossa.	Etäopetus (verkon välityksellä)

Verkko-opetuksen kustannukset muodostuvat työntekijäkuluista sekä tiloista, laitteista ja ohjelmistoista muodostuvista kustannuksista (Ojala 2004). Tilojen, laitteiden ja ohjelmistojen suhteen voidaan tehdä niin halpoja kuin kalliitakin ratkaisuja. On kuitenkin todennäköistä, että merkittävin osa opintojakson kustannuksista muodostuu työntekijäkuluista. Kustannusten tarkempi sijoittuminen opintojakson työvaiheisiin riippuu opintojakson tyyppistä: Materiaalipankeissa ja monimuoto-opetuksessa suurin osa resursseista kuluu opintojakson suunnitteluun, materiaalin tuottamiseen ja opintojakson ulkoasun tekemiseen, verkko-opetuksessa hallinnointiin, opiskelijoiden ohjaukseen ja teknisten ongelmien ratkaisemiseen (ibid.). Verkko-opetuksen kustannuksia on mahdollista laskea esimerkiksi käyttämällä oppimateriaalia uudelleen tai käyttämällä oppimisalustoja, joilla opettajan ei tarvitse itse huolehtia materiaalin esitysmuodosta, kuten HTML-merkkauksesta.

Opettajan työ määrä voi vähentyä, jos verkko-opetusjärjestelmä hoitaa osan mekaanisesta työstä, kuten opiskelijoiden hallinnan ja harjoitusten tarkastuksen. Tässä työssä käsi-

teltävät järjestelmät pystyvät tuottamaan opiskelijoille myös henkilökohtaista, räätälöityä palautetta, joka voi olla jopa kattavampaa kuin opettajan tavallisesti antama. Samalla järjestelmä pystyy välittämään opettajalle yksityiskohtaista tietoa siitä, kuinka opiskelijat ratkaisevat tehtäviä. Verkko-opetus voi siis muuttaa opettajan roolia: Opiskelijoiden vastausten tarkastamisen asemesta opettajat keskittyvät ehkä tulevaisuudessa enemmän niiden analysointiin ja pyrkivät käyttämään näin saatua tietoa opetuksen kehittämiseen.

Verkko-opetuksen rooli yliopisto-opetuksessa kasvaa jatkuvasti. Myös Tampereen teknillinen yliopisto on linjannut, että laitosten on hyödynnettävä tieto- ja viestintäteknikkaa monipuolisesti opetusta ja kehitettävä monimuotoista opetusta, jossa tieto- ja viestintäteknikkaa hyödynnetään pedagogisesti perustellulla tavalla. Erikseen on mainittu oppimisalustojen, eSovellusten, videoteknologian, simulointien, mallinnuksien ja sähköisten aineistojen käyttö. Opettajien on myös kerättävä palautetta opiskelijoilta tieto- ja viestintäteknikan opetuskäytöstä. (Tampereen teknillinen yliopisto 2009)

## 2.2 Matematiikan tehtävät verkko-opetuksessa

Suuri osa verkko-opetukseen liittyvästä kirjallisuudesta ja tutkimuksesta on riippumattomana opetettavasta aineesta. Keränen ja Penttinen mainitsevat verkko-opetuksen työkaluiksi keskustelualueet, chatit, sähköpostin, tehtävät, testit, sanastot, wikit, blogit, tiedotteet, kalenterit ja syötteet (Keränen & Penttinen 2007). Allan lisää mukaan vielä äänitiedostot, sähköpostilistat, uutisryhmät, äänestykset, lomakkeet, videokonferenssit, pikaviestit, verkkopuhelut, podcastit, sosiaalisen median, videoleikkeet, virtuaalikierrokset ja virtuaalimaailmat (Allan 2007). Suurin osa näistä työkaluista on tarkoitettu tekstimuotoisen materiaalin jakamiseen ja työstämiseen, ja soveltuu näin erinomaisesti kieliiin ja reaaliaineisiin, joissa esseiden kirjoittaminen on tavallinen opiskelumuoto.

Matematiikkaa ei kuitenkaan yleensä opiskella kirjoittamalla esseitä. Siksi verkko-opetuksessakin on luontevampaa pyytää opiskelijaa ratkaisemaan laskutehtäviä kuin kirjoittamaan tietyn funktion ominaisuuksista.

Matemaattinen laskutehtävä ja reaaliaineen essee eroavat toisistaan myös arvostelun puolesta. Jälkimmäisen luotettavaan arviointiin pystyy vain aiheeseen perehtynyt ihminen, jolloin järjestelmän tärkeimmiksi tehtäviksi jäävät tehtävänannon välittäminen opettajalta opiskelijoille ja vastausten välittäminen opiskelijoilta opettajalle. Laskutehtäviin on yleensä olemassa yksi tai useampia oikeita vaihtoehtoja, jotka on mahdollista löytää koneellisesti. Matematiikanopetuksen järjestelmältä voidaan siis vaatia erilaista toiminnallisuutta kuin reaaliaineiden opetukseen suunnatulta.

Moni verkko-oppimisen työkaluista perustuu opiskelijoiden väliseen kommunikaatioon, ja niitä on mahdollista hyödyntää myös matematiikanopetuksessa. Tavallinen matema-

tiikan tehtävä on kuitenkin laskuharjoitus, jonka opiskelija suorittaa itsenäisesti. Opiskelijoiden yhteistyötä tukevia matematiikan verkko-opetuksen järjestelmiä ei toistaiseksi ole. Tässä diplomityössä oletetaan opiskelijoiden ratkaisevan matematiikan tehtävänsä yksin.

Yleinen verkko-opetuksen kirjallisuus ei kuitenkaan yleensä mainitse matemaattisia aineita. Yliopisto-opettajat voivat jopa kokea, ettei matematiikanopetus ole mahdollista ilman lähiopetusta (Pirttimäki 2004). Matematiikka ei ole laajasti edustettuna myöskään valtion tuottamassa, vapaasti jaettavassa verkko-oppimateriaalissa: Opetushallitus tarjoaa peruskoulun oppilaille kahdeksan tehtävää ja matematiikan sanastot seitsemälle kielelle, lukion oppilaille tilastomatematiikan kokonaisuuden sekä kaikille peruskoulun, lukion ja ammatillisten oppilaitosten oppilaille soveltuvat 12 tehtävää (Opetushallitus 2012). Tehtävät hyödyntävät Java-, Flash-, PDF- ja JavaScript-tekniikoita sekä Microsoft Internet Explorer ja Microsoft Excel -ohjelmia, joista yhtäkään ei ole suunnattu matematiikanopetukseen.

Sekä verkkokeskustelu matematiikasta että yksinkertaisten pelinomaisten tehtävien ratkaiseminen voivat olla hyödyllinen osa opetusta. On kuitenkin olemassa myös juuri matematiikan opiskeluun tarkoitettuja oppimisympäristöjä, jotka tarjoavat monipuolisempia työkaluja algebrallisten tehtävien jakamiseen, palautukseen ja tarkastukseen.

Verkkotehtävien toiminnallisuus voidaan jakaa kolmeen osaan:

1. tehtävänanto
2. vastauksen palautus ja
3. vastauksen tarkastus.

Järjestelmät toteuttavat nämä toiminnot yleensä kumulatiivisesti. Niinpä järjestelmätkin voidaan jakaa kolmeen kategoriaan toiminnallisuuden perusteella:

1. järjestelmät, joilla jaetaan tehtäviä
2. järjestelmät, joilla jaetaan tehtäviä ja otetaan vastaan vastauksia, ja
3. järjestelmät, joilla jaetaan tehtäviä ja otetaan vastaan ja tarkastetaan vastauksia.

Tehtävien jakaminen verkossa on toiminnoista yksinkertaisin. Matematiikan tehtäviä voidaan jakaa minkä tahansa verkkosivun kautta. Tavallisimmat esitysmuodot ovat teksti, rakenteinen teksti ja kuva, mutta tehtävä voitaisiin tarvittaessa esittää missä tahansa verkon kautta välittyvässä muodossa, esimerkiksi äänitiedostona. Matemaattisten merkintöjen esittäminen selaimessa on kuitenkin merkittävä haaste matematiikan verkko-opetuksessa. Käytössä on monia kilpailevia menetelmiä, jotka perustuvat yleensä kuvien, HTML-elementtien ja JavaScript-skriptikielen käyttöön. Ehdotetuista standardeista suurimman suosion on saavuttanut MathML, jota tukee kuitenkin vasta Firefox-selain. Tehtävät voidaan palauttaa opettajalle kirjallisesti, joko henkilökohtaisesti tai postitse,

tai suullisesti harjoitustilaisuudessa. Opettaja tarkastaa tehtävät ja voi antaa opiskelijalle palautetta lähiopetustilanteessa, verkossa tai muulla haluamallaan tavalla.

Järjestelmä voi sisältää myös sisäänrakennetun palautusmekanismin, joka on tavallisesti HTML-lomake. HTTP-protokolla mahdollistaa sekä tekstin että minkä tahansa tiedoston siirtämisen verkossa. Opiskelija voi siis palauttaa vastauksensa tarvittaessa vaikkapa kuvatiedostona. Hyväksyttävien vastausmuotojen päättäminen ei ole aivan triviaalia, sillä yksinkertaistenkin matemaattisten kaavojen kirjoittaminen tietokoneella voi olla haastavaa opiskelijalle, joka ei omista siihen tarkoitettuja ohjelmistoja. Opettaja tarkastaa tehtävät ja voi antaa opiskelijalle palautetta lähiopetustilanteessa, verkossa tai muulla haluamallaan tavalla.

Tämä diplomityö käsittelee järjestelmiä, jotka tarkastavat opiskelijan vastaukset itsenäisesti. Tällöin vastauksen muodon tärkeys korostuu entisestään. Perinteisessä lähiopetustilanteessa opettaja pystyy toteamaan opiskelijan osaamisen, vaikka vastaus esitettäisiin epätäsmällisessä muodossa. Koneellisessa tarkastuksessa tämä ei kuitenkaan ole mahdollista.

Oppimisympäristöjen tavalliset automaattisesti tarkastettavat vastausmuodot ovat teksti, numero ja monivalinta. Niiden koneellinen tarkastaminen on helppoa.

Tekstimuotoinen vastaus voidaan tarkastaa merkkijonovertailulla. On kuitenkin huomattava, että näin kannattaa tarkastaa lähinnä yksittäisiä sanoja, sillä yhdenkin merkin ero tekee vastauksesta väärän. Tekstimuotoisia vastauksia voidaan matematiikanopetuksessa käyttää esimerkiksi käsitteiden käytön harjoittelemiseen. Tekstikenttään on mahdollista syöttää myös esimerkiksi yksinkertainen yhtälö, mutta tällöin opiskelijaa on erikseen ohjeistettava vastauksen syntaksista: ” $f(x)=2x+1$ ” tulkitaan eri tulokseksi kuin ” $f(x) = 2*x + 1$ ”.

Numeromuotoinen vastaus voidaan tarkastaa vertailuoperaatiolla. Niillä voidaan kattaa suuri osa peruskoulun matematiikanopetuksesta ja monia korkeampien asteiden tehtäviä, mutta on myös monia tehtäviä, joiden lopputulosta ei voida esittää yhtenä numerona.

Monivalintatehtävässä opiskelijalle esitetään mahdolliset vastausvaihtoehdot, joista hän valitsee yhden tai useamman oikean vastauksen. Opiskelijan valitsema vaihtoehto koodataan numeroksi, jota verrataan oikealle vastaukselle annettuun numeroon. Oikein/väärin-vastaukset ovat monivalinnan erikoistapaus, jossa vain toinen vaihtoehtoista on oikea. Koska tietokoneen tarkastama vastaus on täysin erillinen opiskelijan näkemästä vastausvaihtoehdosta, monivalintojen vastausvaihtoehtoina voidaan käyttää mitä tahansa sanoista ja kuvista videoihin ja matemaattisiin kaavoihin. Monivalintoja käyttäessä on kuitenkin otettava huomioon, että opiskelijat soveltavat niihin eri ongelmanrat-

kaisutekniikoita kuin muihin tehtäviin. Opiskelija ei yleensä ratkaise tehtävää tavallisin keinoin ja vertaa sitten saamaansa vastausta annettuihin vaihtoehtoihin, vaan hän tutkii ensin vastausvaihtoehtoja ja päätelee sitten, mikä niistä täyttää parhaiten tehtävänannon (Sangwin 2007).

Matematiikan opiskeluun on kehitetty myös erilaisia interaktiivisia sovelluksia. Osa niistä on integroitu oppimisympäristöihin, osa toimii itsenäisinä sovelluksina. Ne on toteutettu plugin-tekniikoilla, kuten Java-appletteina tai Flash-sovelluksina, ja ne ovat mahdollistaneet toiminnallisuutta, johon selaimen perustekniikat eivät aiemmin ole pystyneet<sup>1</sup>. Tyypillinen interaktiivisessa sovelluksessa suoritettava tehtävä on annettujen pisteiden, suorien, kuvaajien tai kuvioden merkitseminen koordinaatistoon. Vastauksen tarkastamiseen ei ole yleistä menetelmää, vaan jokainen sovellus tulkitsee sen oikeellisuuden oman sisäisen logiikkansa avulla. Interaktiiviset sovellukset kurovat umpeen kynä-paperi-tekniikoiden ja verkko-opiskelun välistä kuilua esittämällä tehtävät opiskelijoille tutussa muodossa. Niiden toiminnallisuus on kuitenkin rajoitettua: Yhdellä sovelluksella voi yleensä ratkaista vain yhdenlaisia tehtäviä. Lisäksi opiskelija tarvitsee selaimensa sopivan plugin-laajennuksen. Myöskään interaktiivisten sovellusten toteuttaminen ei ole matematiikanopettajalle triviaali tehtävä, sillä se voi vaatia Java-, ActionScript-, JavaScript- tai HTML5-tekniikoiden perusteellista hallintaa.

Parhaalta vaihtoehdolta vaikuttaakin vastausten esittäminen jollakin matemaattisella syntaksilla. Jokaisella matematiikkaohjelmistolla, kuten Maple, Mathematica, MatLab tai Maxima, on oma syntaksinsa, jolla kirjoitetuilla lausekkeilla ne pystyvät suorittamaan operaatioita. Kun opiskelija syöttää oman vastauksensa matematiikkaohjelmiston käyttämällä syntaksilla, vastaus voidaan siirtää suoraan ohjelmiston tarkastettavaksi.

Muita vaihtoehtoja matemaattisen syntaksin esittämiseen esimerkiksi ovat LaTeX-ladontajärjestelmä tai MathML-merkintäkieli. Niillä esitettyjä lausekkeita ei voida tarkastaa suoraan, mutta ne voidaan muuttaa edelleen matematiikkaohjelmiston ymmärtämään muotoon. Erillisen merkintäkielen valinta voi olla perusteltua, jos eri tehtävät lähetetään eri matematiikkaohjelmistojen tarkastettaviksi. Syntaksin valintaan vaikuttavat myös käytössä olevat syöttötyökalut, opiskelijoiden ikä ja matematiikkaohjelmistojen muu hyödyntäminen opetuksessa. MathML:n kirjoittaminen käsin on harvoin järkevää, alakoulun oppilaan ei voida olettaa käyttävän samanlaisia tieteellisiä merkintöjä kuin yliopisto-opiskelijan, ja teknisten alojen opiskelijat saattavat olla jo muissa yhteyksissä tutustuneet johonkin matematiikkaohjelmistoon ja sen syntaksiin. Esimerkiksi signaalinkäsittelyssä tehdään paljon työtä MatLabilla, jolloin sen käyttö matematiikanopiskelussakin lienee opiskelijoille luontevaa.

---

<sup>1</sup> Selainten nopean kehityksen ansiosta moni näistä sovelluksista olisi jo mahdollista toteuttaa moderneissa selaimissa HTML5- ja JavaScript-tekniikoilla.



Jotkin järjestelmät tarjoavat HTML:n tekstikenttää monipuolisempia syöttötyökaluja matemaattisten merkintöjen syöttämiseen. Graafisella editorilla opiskelija voi valita haluamansa merkinnän, esimerkiksi integraalin, jonka työkalu lisää syöttöalueelle. Sen jälkeen opiskelija voi täydentää kaavaan tarvittavat osat, kuten integraalin rajat ja integroitavan lausekkeen. Syöttötyökalujen käyttö vaikuttaa vain siihen, kuinka opiskelija näkee syöttämänsä lausekkeet, ei järjestelmän käyttämään syntaksiin.

Valitusta syntaksista riippumatta vastausten tarkastaminen matematiikkaohjelmistolla antaa parhaat edellytykset hyvien tehtävien luomiseen. Symbolista laskentaa hyödyntävät ohjelmistot pystyvät käsittelemään opiskelijan syöttämiä lausekkeita huomattavasti muita vaihtoehtoja tehokkaammin. Niiden tarjoamat tarkastustyökalut eivät rajoitu merkkijonovertailuun, vaan pystyvät ottamaan huomioon matematiikan aksioomat ja ominaispiirteet. Matematiikkaohjelmisto ymmärtää, että ” $f(x)=2(\frac{1}{2}+x)$ ” tulkitaan samoin kuin ” $f(x)=2x+1$ ”, ja voi kertoa opiskelijalle, että vastaus on oikea, joskaan ei esitetty parhaassa mahdollisessa muodossa.

Palaute on oppimisen kannalta välttämätöntä. Hattien ja Timperleyn mukaan paras palaute antaa opiskelijalle tietoa tämän tekemistä virheellisistä oletuksista, auttaa häntä löytämään parempia strategioita ja motivoi häntä jatkamaan tehtävän yrittämistä (Hattie & Timperley 2007). Tehtävän tekemisen ja palautteen saamisen väliin jäävällä ajalla on suuri merkitys opiskelijan motivaatiolle. Verkko-opetusjärjestelmien tarjoama reaaliaikainen palaute kannustaa opiskelijaa jatkamaan tehtävän ratkaisemista virheellisen yrityksen jälkeen. Sosa et al. totesivat tilastotieteen opiskelua koskevassa meta-analyyssissään tietokoneavusteista opetusta saaneiden opiskelijoiden menestyneen pelkkää lähiopetusta saaneita paremmin. Tietokoneavusteista opetusta saaneiden opiskelijoiden keskuudesta parhaiten menestyivät ne, jotka saivat järjestelmältä jatkuvaa palautetta, eivätkä vain yhtä kokonaisarvosanaa koko opintojakson suorituksesta. (Sosa et al. 2011) Myös opiskelijat ovat suhtautuneet automaattisesti tarkastettaviin matematiikan tehtäviin positiivisesti (Rasila et al. 2007). Tehtävien automaattisella tarkastuksella on siis suuri merkitys matematiikan verkko-opetuksessa.

### 2.3 Matematiikan verkko-opetusjärjestelmien arviointi

Tehtävien jakaminen ja vastausten palauttaminen ja tarkastaminen ovat matematiikan verkko-opetusjärjestelmän avaintoimintoja. Järjestelmien parempaa arviointia ja vertailua varten on kuitenkin syytä asettaa niille muitakin kriteerejä. Bokhove ja Drijvers asettivat vertailevassa tutkimuksessaan arvioitaville järjestelmille seuraavat ehdot:

- matematiikan tuki
- mahdollisuus tehtävien laatimiseen
- datan säilytys ja
- tekninen tuki.

(Bokhove & Drijvers 2010)

Matematiikan tuki sisältää matemaattisen notaation käytön ja algebrallisten operaatioiden tuen. Matematiikan verkkotehtävien on siis jäljiteltävä perinteisiä kynä-paperimenetelmiä. Näin opiskelijan aloituskynnys verkkotehtävien tekemiseen madaltuu ja eri työkalut tukevat toisiaan.

Mahdollisuus laatia omia tehtäviä on erityisesti yliopisto-opetuksessa tärkeä ominaisuus. Opettajat eivät yleensä seuraa valmista opetussuunnitelmaa, vaan vastaavat omien opintojaksojensa oppimistavoitteiden määrittelystä ja oppimateriaalin valinnasta tai tuottamisesta. Tästä seuraa luonnollisesti, että opettajat haluavat myös laatia käyttämässä tehtävät itse. Järjestelmät, jotka tarjoavat vain mahdollisuuden valita opiskelijoille annettavat tehtävät valmiista tehtäväpankista, saattavat soveltua alemmille kouluasteille, joiden opetussuunnitelmat muuttuvat hitaasti, mutta yliopistoissa niihin ei kannata panostaa. Tehtävien laatimisen on myös oltava kohtuullisen helppoa: Opettajan ei voida olettaa esimerkiksi ohjelmoivan joka tehtävää varten erillistä Java-applettia.

Datan säilytykseen liittyy kaksi kriteeriä. Ensinnäkin opiskelijan on pystyttävä käyttämään järjestelmää aika- ja paikkariippumattomasti. Käytännössä tämä tarkoittaa, että järjestelmään on päästävä Internet-yhteydellä. Käyttöä ei voi rajoittaa esimerkiksi tiettyyn yliopiston tietokonealueeseen. Toiseksi järjestelmän on tallennettava opiskelijoiden tuottama data keskitetysti. Järjestelmä ei siis voi olla ohjelma, jonka opiskelija asentaa koneelleen ja jota hän käyttää ilman Internet-yhteyttä. Datan tallennus yhteen paikkaan mahdollistaa sen tarkastelun jälkikäteen, jolloin opettaja saa arvokasta tietoa opiskelijoiden toiminnasta ja osaamisesta.

Viimeinen kriteeri on, että järjestelmälle tulee olla saatavilla teknistä tukea. Tukipalveluiden tarve riippuu suuresti siitä, onko kyseessä kaupallinen vai avoimen lähdekoodin ohjelmisto, ja mitkä ovat oppilaitoksen omat tekniset valmiudet: teknillisen alan yliopistolla lienee paremmat valmiudet käyttää avoimen lähdekoodin järjestelmää ilman ulkoisia tukipalveluita kuin alakoululla. On myös varmistuttava siitä, että järjestelmää on mahdollista käyttää tulevaisuudessakin. Yhden vuoden lisenssillä myytävä kaupallinen ohjelmisto ei ole yliopistolle realistinen vaihtoehto, mikäli toimittaja ei takaa mahdollisuutta lisenssin jatkamiseen.

Bokhoven ja Drijversin suunnittelema arviointityökalu tarkastelee matematiikan verkko-opetusjärjestelmiä algebran, käytön, arvioinnin ja yleisten tekijöiden osalta. Neljään kategoriaan kuuluu yhteensä 27 arviointikohtaa, joille on määritetty painokertoimet. Työkalun laatimiseen osallistui tutkimuksen tekijöiden lisäksi 27 matematiikanopetuksen, teknologia-avusteisen opetuksen ja/tai arvioinnin asiantuntijaa. (ibid.)

Bokhoven ja Drijversin arviointityökalu antaa suurimmat painokertoimet seuraaville ominaisuuksille:

1. järjestelmän vakaus ja suorituskyky (painokerroin 4,89)
2. järjestelmän helppokäyttöisyys opiskelijan näkökulmasta (painokerroin 4,85)
3. järjestelmän kyky esittää matemaattisia merkintöjä (painokerroin 4,81)
4. järjestelmän matemaattinen oikeellisuus ja uskollisuus perustana oleville matemaattisille periaatteille (painokerroin 4,74) ja
5. järjestelmä tallentaa opiskelijoiden vastaukset (painokerroin 4,70).

Järjestelmän lisenssin tyyppi (avoin/kaupallinen) jää arvostelukriteereistä vähiten merkitsevimmäksi painokertoimella 3,41. Aalto-yliopistolla tehdyssä matematiikan verkko-opetusjärjestelmien arvioinnissa lisenssin tyyppi nousi kuitenkin erittäin tärkeäksi kriteeriksi, sillä avoimen lähdekoodin järjestelmän käyttö mahdollistaa omat muutokset ja parannukset (Rasila et al. 2007). Aalto-yliopiston mukaan olemassa olevien järjestelmien suurimmat puutteet ovat suurten opintojaksojen käyttäjänhallinta, suomen kielen tuki ja yhteensopivuus eri alustojen kanssa (ibid.). Näistä Bokhoven ja Drijversin arviointityökalu mainitsee vain kielten tuen painoarvolla 4,11.

Tässä diplomityössä kiinnitetään huomiota erityisesti opettajan työkalujen monipuolisuuteen ja käytettävyyteen. Bokhoven ja Drijversin arviointityökalussa niitä ei pidetä yhtä mielenkiintoisina: Opettajan mahdollisuus luoda omia tehtäviä on arvioitu 14. tärkeimmäksi painokertoimella 4,33, vähemmän tärkeäksi kuin 11. sijalle noussut valmiin materiaalin saatavuus (painokerroin 4,48). Opiskelijan käyttöliittymän käytettävyys on kriteereistä peräti toiseksi tärkein, opettajan käyttöliittymän käytettävyys ei kuulu lainkaan kriteereihin. Vaikuttaakin siltä, että Bokhove ja Drijvers arvioivat matematiikan verkko-opetusjärjestelmiä ensisijaisesti oppimistulosten perusteella ottamatta juurikaan huomioon matematiikanopettajien näkökulmaa. Opetusta järjestävää organisaatiota kiinnostavat ominaisuudet, kuten järjestelmän hinta ja lisenssin tyyppi, on arvioitu vähän merkitseviksi.

Tässä diplomityössä esitellään tarkemmin avoimen lähdekoodin STACK (luku 3) sekä MathBridge-matematiikanoppimisympäristö tehtävämootoreineen (luku 2.5). Muita järjestelmiä ei työn puitteissa testattu yksityiskohtaisesti tai vertailtu. Lähdekirjallisuudesta ja muista yhteyksistä nousi esiin myös muita järjestelmiä, jotka voisivat sopia yliopistotasoiseen matematiikanopetukseen.

**Maple T.A.** (MapleSoft 2012) on Maple-matematiikkaohjelmiston luojan kehittämä maksullinen tehtävä- ja tenttijärjestelmä. Opettajat voivat luoda järjestelmässä omia tehtäviä tai käyttää valmiita tehtäviä, joita on saatavilla sekä ilmaiseksi että lisämaksusta. Valmiin sisällön joukosta löytyy jopa kokonaisia opintojaksoja, joiden sisällöt ovat melko samankaltaisia Tampereen teknillisellä yliopistolla opettettavien kanssa. Omien tehtävien luominen tapahtuu lomakepohjaisessa käyttöliittymässä ja edellyttää opettajalta vain Maplen syntaksin tuntemista. Opettajan aloituskynnys järjestelmän käyttöön on siis hyvin matala.

Kuten kaupallisille järjestelmille on tyypillistä, Maple T.A.:n ulkoasu näyttää avoimen lähdekoodin järjestelmiä viimeistellymmältä. Opiskelija voi syöttää matemaattisia merkintöjä sisältävät vastauksensa Maple-syntaksin lisäksi myös kaavaeditorilla. Järjestelmä sisältää paljon ominaisuuksia, joista erityisen mielenkiintoinen on mahdollisuus muuttaa tehtävänantoa opiskelijan vastauksen perusteella: Väärän vastauksen jälkeen voidaan näyttää sama tehtävä helpotetuilla alkuarvoilla, helpotettu tehtävä tai sama tehtävä vaiheistettuna.

Maple T.A.:lla on useita lisensointivaihtoehtoja. Järjestelmälle voidaan ostaa ylläpito-palvelu MapleSoftilta, tai se voidaan asentaa omalle palvelimelle, jolloin maksetaan vain lisenssistä ja mahdollisesti ohjelmiston päivityksistä. Opetuksen järjestäjä voi ostaa lisenssin kaikille opiskelijoille tai kehottaa tehtävistä kiinnostuneita opiskelijoita ostamaan henkilökohtaiset lisenssit. Maksullisen järjestelmän käyttöönoton suurin ongelma on päätöksentekokynnys: Järjestelmän perinpohjainen testaaminen ennen ostopäätöstä on vaikeaa, ja lisenssit kannustavat pitkäaikaiseen sitoutumiseen. Maksullinen järjestelmä ei kuitenkaan ole välttämättä kallein vaihtoehto. Ilmainen avoimen lähdekoodin järjestelmä voi vaatia paljon kehitys-, mukautus- ja ylläpitotyötä, joka näkyy työntekijäkuluissa. Olisikin mielenkiintoista vertailla Maple T.A.:n kaltaisen maksullisen ja ilmaisen järjestelmän todellisia kustannuksia. Maple T.A.:n ja STACKin muista eroista kiinnostuneita suositellaan tutustumaan Blåfieldin pro gradu –työhön (Blåfield 2009).

**MathXL** (Pearson Education 2012) on oppikirjakustantaja Pearson Educationin kehittämä kaupallinen järjestelmä. Se tarjoaa kokoelman tehtäviä, jotka liittyvät tiettyihin kustantajan julkaisemiin oppikirjoihin. Opettajien on mahdollista luoda järjestelmällä myös omia tehtäviään. MathXL:n käyttö edellyttää opiskelijoilta Pearson Educationin julkaiseman oppikirjan tai erillisen MathXL-lisenssin ostamista. Lisenssi on voimassa 12 kuukautta. Tampereen teknillisen yliopiston matematiikan opintojaksot eivät pääsääntöisesti perustu yhteen oppikirjaan, joten MathXL:n hyödyt ovat kyseenalaiset. Järjestelmästä ei ole saatavilla kattavaa dokumentaatiota eikä kokeiluersiota.

**WeBWorK** (Mathematical Association of America) on ilmainen avoimen lähdekoodin järjestelmä. Se on suunnattu erityisesti yhdysvaltalaisen yliopistojen kandidaatin tutkin-tojen matematiikanopetukseen ja sisältää yli kahdenkymmenen tuhannen valmiin tehtävän kokoelman. WeBWorK ei käytä erillistä matematiikkaohjelmistoa, vaan vastausten tarkastamiseen käytetään Perl-ohjelmointikielen pohjautuvalla Problem Generating - kielellä (PG) kirjoitettuja algoritmeja. Omien tehtävien luominen edellyttää opettajalta ohjelmointitaitoa, mutta jo valmiiden tehtävien käyttäminen voisi tuoda opetukseen selvää lisäarvoa hyvin pienillä kustannuksilla. WeBWorK tarjoaa myös mahdollisuuden maksulliseen ylläpitopalveluun.

Lähdekirjallisuus esitteli myös muita vaihtoehtoja, joita ei pidetä sopivina Tampereen teknillisen yliopiston opetuskäyttöön. Bokhoven ja Drijversin vertailussa parhaaksi arvioitu **Digital Math Environment (DME)** (Freudenthal Institute 2004) ei sovellu suomalaisen yliopistoon, sillä se on saatavilla vain Alankomaissa ja Belgiassa. Lisäksi järjestelmä on suunniteltu toisen asteen matematiikanopetukseen 12–18-vuotiaille oppilaille, mikä antaa aiheita epäillä, ettei sillä ole mahdollista esittää ja arvioida kaikkia yliopistomatematiikan aihealueita. STACKin esikuvana toiminutta **AiMia** (University of York) ei pidetä varteenotettavana vaihtoehtona, sillä sen kehitys on lopetettu. Viimeisin versio julkaistiin vuonna 2003, mikä antaa syytä epäillä, ettei järjestelmä ole enää yhteensopiva muiden käytössä olevien ohjelmistojen kanssa tai tue uusia selaintekniikoita. **WWW Interactive Multipurpose Serverillä (WIMS)** (Xiao) on toteutettu monipuolisesti erilaisia tehtäviä erityisesti ranskan kielellä. Se ei kuitenkaan käytä eikä tue matemaattisia merkintöjä. WIMS käyttää tehtävän kuvailuun Online Exercise Format -merkintäkieltä (OEF). Muita OEF:ää käyttäviä järjestelmiä ei tiedettävästi ole.

## 2.4 Matematiikan verkko-opetus Tampereen teknillisellä yliopistolla

Tampereen teknillisen yliopiston matematiikanopetus on pääsääntöisesti lähiopetusta. Tyypillinen matematiikan opintojakso koostuu luennoista ja harjoitustilaisuuksista, joissa opiskelijat esittävät ratkaisuja etukäteen jaettuihin laskutehtäviin ja/tai ratkovat tehtäviä opettajan ohjauksessa. Osaan matematiikan opintojaksoista sisältyy myös tietokoneharjoitustilaisuuksia, joissa opiskelijat ratkovat laskutehtäviä yliopiston tietokoneilla esimerkiksi MatLab- tai Maple-ohjelmistoilla. Internetiä käytetään opintojaksoille ilmoittautumisiin, opintojakson järjestelyistä tiedottamiseen ja oppimateriaalin ja harjoitustehtävien jakamiseen. Opettaja voi valita alustaksi TTY:n intranetin, erillisen kotisivun tai oppimisympäristön Moodlen.

Opiskelijat ilmoittautuvat kaikille TTY:n järjestämille opintojaksoille ja tentteihin intranet POPin kautta. Opettajien on lisättävä järjestelmään vuosittain tieto järjestämistään kursseista ajankohtineen ja sisältökuvauksineen. Opettaja voi halutessaan lisätä opintojakson intranet-sivulle myös enemmän tietoa suoritusvaatimuksista ja käytännöistä, tiedottaa muutoksista opintojaksolle ilmoittautuneille opiskelijoille ja jakaa opintojaksoon liittyviä tiedostoja. Osa opettajista ei jaa tietoa opintojaksoistaan muualla verkossa.

Opettaja voi halutessaan tehdä opintojaksolle myös oman kotisivun laitoksen tarjoamaan sivutilaan. Opintojaksojen kotisivut ovat tavallisesti lyhyitä ja yksinkertaisia, ja ne sisältävät tietoa aikataulusta ja käytännön järjestelyistä sekä oppimateriaalia ja tehtäviä PDF-dokumentteina. Jaettavan materiaalin määrä jää opettajan oman harkinnan varaan. Kotisivun tekeminen edellyttää, että opettaja hallitsee HTML-merkintäkielen ja verkkojulkaisemisen perusteet. TTY suosittelee opettajille intranetin ja Moodlen käyttöä.

Moodle on TTY:n virallinen oppimisympäristö. Sen avulla voidaan tarjota opiskelijoille tietoa opintojaksosta, tiedottaa osallistujille muutoksista ja jakaa materiaalia. Lisäksi Moodle sisältää paljon erilaisia työkaluja muun muassa tehtävien palautukseen, chat- ja foorumikeskusteluihin, kyselyihin ja sisällön sosiaaliseen tuottamiseen. Moodlea ei tätä diplomityötä kirjoittaessa käytetä laajasti matematiikanopetuksessa, mutta tilanne muuttune lähivuosina.

Korhonen ja Pantzar tyypittelevät verkko-opettajat kokemustaustan mukaan neljään kategoriaan: kokeneisiin verkkopedagogeihin, hands on -osaajiin, käytännön kokeilijoihin ja tiedonjanoisiin noviiseihin (Korhonen & Pantzar 2004). Tampereen teknillisen yliopiston matematiikanopettajat sijoittuvat pääosin kahteen viimeiseen.

TTY:llä käytetään tietokoneavusteista tehtävien tarkastusta perustaitojen testissä ja matematiikkajumppassa. Perustaitojen testi on lukion matematiikan pitkään oppimäärään perustuva tietokoneella tehtävä harjoitus. Syksyllä 2012 testin suorittivat kaikki TTY:llä aloittaneet. Opiskelijat, jotka eivät saa perustaitojen testistä riittävästi pisteitä, joutuvat suorittamaan matematiikkajumpan, joka koostuu lukion pitkän matematiikan sisältöjä kertaavista tehtävistä.

Perustaitojen testi on toteutettu selainpohjaisena sovelluksena. Ohjelma esittää opiskelijalle tehtävänannon sekä huolehtii vastausten näyttämisestä ja tallentamisesta. Aiemmin opiskelijan syöttämät vastaukset lähetettiin CGI-tekniikalla toiselle palvelimelle, jolla niitä käsiteltiin Maple-matematiikkaohjelmiston palvelinversio MathEdgellä ja joka palautti tuloksen. (Nykänen & Pohjolainen 1998, Nykänen 2000, Pohjolainen et al. 2006) Testi kattoi aihealueet lukuarvot, lausekkeet, yhtälöt, epäyhtälöt, logaritmit, potenssit, trigonometria, derivointi ja integrointi ja koostui 16 tehtävästä, joiden lähtöarvot arvottiin joka opiskelijalle erikseen (Huikkola et al. 2008). Myöhemmin perustaitojen testit käännettiin Maxima-matematiikkaohjelmistolla tarkastettaviksi.

Syksyllä 2010 jumppa järjestettiin STACK-ohjelmistoa käyttäen. Migraatioon saatiin apua Aalto-yliopistolta, joka antoi TTY:n käyttöön STACKin ensimmäisestä versiohaarasta kehittämänsä version sekä käänsi matematiikkajumpan tehtävät STACKin käyttämään XML-muotoon.

Matematiikkajumppa aloitti osana Insinöörimatematiikka 1 -opintojaksoa. Perustaitojen testissä heikosti menestyneet opiskelijat suorittivat lukion matematiikan pitkää oppimäärää kertaavia lisätehtäviä tietokoneharjoituksina. Harjoitukset oli mahdollista suorittaa joko pienryhmäharjoituksissa tai itsenäisesti (Pohjolainen et al. 2006). Jumppatehtävät toteutettiin teknisesti samoin kuin perustaitojen testi. Lisäksi A&O-oppimisympäristöä käytettiin opiskelijoiden hallintaan (ibid.).

Syksyllä 2011 matematiikkajumppaan saivat osallistua myös perustaitojen testin hyväksytysti suorittaneet. Matematiikkajumpan tehtävät ja oppimateriaalit sijaitsivat MathBridge-oppimisolustalla, jota laajennettiin tukemaan tehtävien automaattista tarkastusta STACKiin pohjautuvalla tekniikalla, ja tiedotus tapahtui TTY:n sosiaalisen median sivusto TTY-Piirin kautta (Matematiikkajumppa 2011).

## 2.5 MathBridge

MathBridge on Euroopan unionin rahoittama eContentplus-ohjelmaan kuuluva projekti, johon osallistui vuosina 2009–2012 yhdeksän eurooppalaista yliopistoa seitsemästä eri maasta. Koordinaattorina toimi saksalainen Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI). Projektin tavoite oli tuottaa monikielisiä matematiikan oppimisympäristöitä, jotka kurovat umpeen kuilua lukio- ja yliopistomatematiikan välillä. Sisältöön kuuluu muun muassa teoreemoja, todistuksia, määritelmiä, esimerkkejä ja interaktiivisia tehtäviä. (MathBridge 2010)

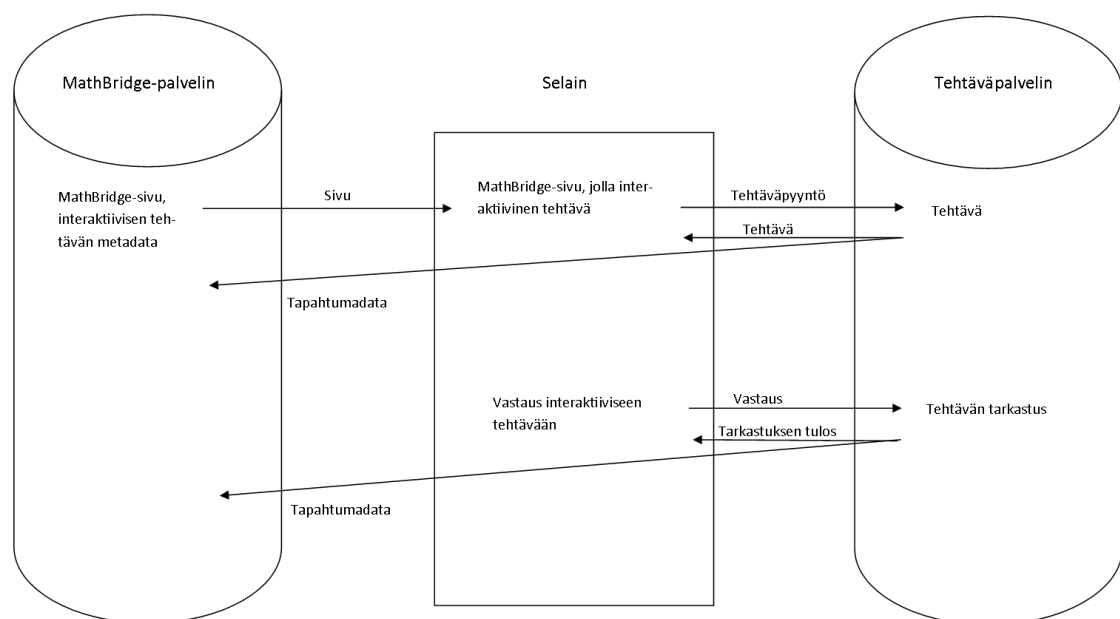
Projektin päätuotos oli MathBridge-oppimisympäristö, jossa käyttäjät voivat tutustua valmiisiin matematiikan kursseihin tai räätälöidä omiin tarpeisiinsa soveltuvia kursseja oppimisympäristöön kokoamalla. MathBridgellä ei ole keskitettyä ylläpitoa, vaan jokainen projektiin osallistuva yliopisto asentaa ohjelman omille palvelimilleen. Osallistujat saavat myös päättää itse, mitä osia MathBridge-sisällöistä ottaa käyttöön.

Tampereen teknillinen yliopisto toimi konsortion jäsenenä koko projektin ajan. TTY tuotti MathBridge-oppimisympäristöön TUT-sisältökokoelman, joka sisälsi Johdatus korkeakoulumatematiikkaan ja Insinöörimatematiikka 1-4 -kurssien materiaalin sekä joukon interaktiivisia STACK-tehtäviä (78 kpl). Materiaalista tuotettiin suomen- ja englanninkieliset versiot, jotka muut partneriyliopistot käänsivät myös espanjaksi, hollanniksi, ranskaksi, saksaksi ja unkariksi. Lisäksi TTY käänsi suomeksi LeAM-Calculus-sisältökokoelman sekä osia muista sisältökokoelmista. (Pohjolainen et al. 2012)

MathBridgen pohjana toimii matematiikan verkko-opiskeluun luotu ActiveMath-oppimisympäristö, jonka ovat kehittäneet Deutsches Forschungszentrum für Künstliche Intelligenz ja Saarland University. ActiveMath on erityisesti matematiikan opiskeluun tarkoitettu web-pohjainen, monikielinen järjestelmä (Deutsches Forschungszentrum für Künstliche Intelligenz & Saarland University 2007). Sen palvelinvaatimukset ovat Java, Tomcat, Apache ja Derby-tietokanta. Selaimelta ActiveMath vaatii tuen JavaScriptille ja MathML-kuvailukielelle. Koska selaimet eivät vielä juurikaan tue MathML:ää, ActiveMathia suositellaan käytettävän vain Mozilla Firefoxilla.

Tampereen teknillinen yliopisto oli mukana suunnittelemassa MathBridgeen rajapintaa, joka mahdollistaa interaktiivisten tehtävien tekemisen toisella palvelimella. Kuva 2.1 esittää, kuinka opiskelijan selain, MathBridge-palvelin ja tehtäväpalvelin kommunikoi-

vat opiskelijan ratkaistessa interaktiivista tehtävää. MathBridge-ohjelma sisältää metadataa tehtävistä, muttei itse tehtäviä. Kun opiskelija avaa tehtävän sisältävän sivun, MathBridge-palvelin lähettää hänen selaimensa sivun muun sisällön sekä tiedon tehtävän osoitteesta. Selain ottaa yhteyden tehtäväpalvelimeen, joka liittyy tehtävän sisällön sivulle sisäiseen kehykseen (HTML-elementti <iframe>). Samanaikaisesti tehtäväpalvelin lähettää MathBridge-palvelimelle tiedon opiskelijan suorittamasta toimenpiteestä. Opiskelijan vastatessa tehtävään vastaus lähetetään tarkastettavaksi tehtäväpalvelimelle, joka lähettää tarkastuksen tuloksen sekä opiskelijalle että MathBridge-palvelimelle. Selaimen ja palvelinten kommunikointi tapahtuu opiskelijan huomaamatta: Hän ei missään vaiheessa huomaa, ettei kaikki sivujen sisältö ole peräisin samasta lähteestä, ellei hän erikseen tarkastele sivujen lähdekoodia.



**Kuva 2.1.** MathBridge-tehtävän suoritukseen liittyvä datansiirto.

Datansiirto on toteutettu XML-RPC-protokollalla HTTP:n yli. Huomattava on, että koska osa sivun sisällöstä tuodaan toiselta palvelimelta, jotkin selaimet eivät same origin -periaatteen mukaisesti käsittele sitä JavaScriptilla. Huomattavin vaikutus on, ettei MathBridgen interaktiivisia tehtäviä ole mahdollista ratkaista esimerkiksi Google Chrome -selaimella. Ongelma voitaisiin välttää sijoittamalla MathBridge-ohjelmisto ja tehtävämoottori samalle palvelimelle. Toistaiseksi niin ei ole tehty ohjelmien erilaisiin tietoturva vaatimuksiin liittyvistä syistä.

Tehtäväpalvelimen ohjelma huolehtii tehtävien säilytyksestä, esittämisestä opiskelijalle ja tarkastamisesta. Se ei sisällä käyttäjätunnistusta tai vastausten tai tarkastusten tulosten tallennusta. MathBridge-ohjelmisto hoitaa nämä tehtävät tehtävämoottorilta saamansa tapahtumadatan perusteella. Tapahtumadatan oikean alkuperän varmistamiseksi tehtävämoottori käyttää MathBridgen välittämää tunnistetta. Tehtävämoottoriin ei kuulu



myöskään käyttöliittymää tehtävien luomiselle. Toteutus ei ota kantaa siihen, kuinka tehtävät on luotu. Ne on vain tallennettava yhteensopivina XML-tiedostoina.

MathBridgen tehtävämoottori on STACKiin pohjautuva, Maxima-matematiikkaohjelmistoa hyödyntävä kevyt sovellus. Sen käyttämät tehtävät on luotu Aalto-yliopiston STACKin ensimmäisestä versiohaarasta kehittämällä OtaSTACKilla. Tehtävämoottori ei kuitenkaan toteuta kaikkea OtaSTACKin toiminnallisuutta. Tehtävän kuvaus onkin vain alijoukko OtaSTACK-tehtävän kuvauksesta. Merkittävimmät erot OtaSTACKin tarjoamiin tehtäviin ovat seuraavat:

- Tehtävämoottori ei esitä kuvia.
- Tehtävämoottori tarkastaa vain, onko vastaus oikein vai väärin.
- Tehtävämoottori tarkastaa vastauksen yksivaiheisesti, vastauspuuta ei ole.
- Tehtävämoottori esittää vain yksittäisiä tehtäviä, ei tehtäväsarjoja.

MathBridgeä käytettiin Tampereen teknillisellä yliopistolla ensimmäisen kerran syksyllä 2011 matematiikkajumpassa (luku 2.4). Kuvassa 2.2 on esimerkki matematiikkajumpan tehtävästä MathBridgessä. Tämän diplomityön kirjoitusaikana keväällä 2012 matematiikkajumpasta kerätyn datan jalostus ja tutkimus ovat vielä kesken, eikä ole tiedossa, poikkeavatko MathBridgessä suoritettujen matematiikkajumpan tulokset aiempien, ilman automaattisesti arvioituja tehtäviä järjestettyjen toteutuskertojen toteutuksista. On kuitenkin odotettavissa, että MathBridgestä kerätty data antaa lisätietoa siitä, kuinka heikoimmat opiskelijat ratkaisevat matematiikan tehtäviä.

The screenshot shows the MathBridge web application interface. At the top left is the MathBridge logo with 'Beta' and 'MATH-BRIDGE EDUCATION SOLUTION'. The main header area contains 'Logaritmit - Jumppatehtävät' and 'Oppimateriaali - x'. The left sidebar shows a navigation menu with 'Logaritmit' selected. The main content area displays a problem titled 'Logaritmi-epäyhtälö' with the equation  $\ln(x) < 12$ . Below the equation, there is a text prompt: 'Anna vastauksesi muodossa "x<a" tai "x>a", missä a on juuren tarkka arvo. Erisuuruusmerkit merkitään <, >.' A note below says: 'Huom! Vastauksessasi ln() merkitsee luonnollista logaritmia, esimerkiksi ln(32). Luonnollisen logaritmin kantaluku e merkitään e.' There is an input field for the answer, a 'Tarkista vastauksesi synteksi' button, and a 'Sulje tehtävä' button. At the bottom, there are links for 'Näytä malliratkaisu' and 'Oppimateriaali'.

**Kuva 2.2.** MathBridge-tehtävä.

MathBridge tallentaa tapahtumatietoja opiskelijan suorittamista toiminnoista, kuten kirjautumisista ja sivulatauksista. Myös tehtävämoottorin lähettämä tieto tehtävien aloittamisesta, ratkaisuyrityksistä ja tuloksista tallentuu MathBridgen tapahtumalokiin. Tapahtumia voidaan tarkastella MathBridgen raportointityökalun avulla. Raporteista on mahdollista tehdä hakuja esimerkiksi tietyltä aikaväliltä tai tiettyyn opiskelijaan kohdistuen, ja niistä voidaan tarkastella kaikkia tapahtumia tai valmiita koosteita.

MathBridgen raportointityökalut eivät kuitenkaan anna täysin luotettavaa dataa tehtävämootorilla tehdyistä tehtävistä. MathBridge-ohjelmisto tallentaa osan tehtävämootorin lähettämästä datasta useampaan kertaan, mikä vääristää tilastoja. Tämän takia esimerkiksi koosteraporttien tehtävämäärät ja onnistumisprosentit ovat virheellisiä. Virheet ovat kuitenkin systemaattisia, joten ne on mahdollista poistaa raporteista ohjelmallisesti.

MathBridge-projekti päättyi tammikuussa 2012. Tampereen teknillisellä yliopistolla ei toistaiseksi ole tehty päätöksiä MathBridgen tehtävämootorin kehittämisestä tai MathBridgeä käyttöönotosta muilla opintojaksoilla. MathBridgen rajoittavimpana tekijänä pidetään ActiveMath-alustaa, joka mahdollistaa olemassa olevien sisältöjen tehokkaan hyödyntämisen, muttei juuri uuden sisällön luomista. MathBridgeä varten luodun tehtävämootorin kehittämisen lisäksi muita mahdollisia vaihtoehtoja ovat uuden tehtävämootorin ja oppimisympäristön käyttöönotto ja MathBridgen interaktiivisten tehtävien siirtäminen ActiveMathista toiseen oppimisympäristöön.

## 2.6 Vaatimukset verkko-opetusjärjestelmälle

Yksi työn tavoitteista on arvioida luvussa 3 esiteltävän STACK-järjestelmän soveltuvuutta matematiikan verkko-opetuskäyttöön Tampereen teknillisellä yliopistolla. Jotta järjestelmän käyttöönottoa voidaan suositella, sen on täytettävä seuraavat vaatimukset:

- 1) yliopistotasoiset tehtävät
- 2) riittävä suorituskyky ja tekninen toimivuus
- 3) opiskelijoiden hyväksyntä
- 4) mahdollisuus uusien tehtävien laatimiseen
- 5) vastausten ja arvostelujen tallennus
- 6) hyvä hinta-laatu-suhde ja
- 7) integroituvuus muihin käytössä oleviin järjestelmiin.

Vaatimukset perustuvat sekä luvussa 2.3 esiteltyihin yleisiin arviointikriteereihin että TTY:n Matematiikan laitoksen aiempiin kokemuksiin tehtävien automaattisesta arvioinnista.

Vaatus tehtävien yliopistotasaisuudesta sisältää matemaattisten merkintöjen esityksen tehtävänannossa. Järjestelmän tarkastustoiminnon on pystyttävä arvioimaan vastauksina ei-triviaaleja matemaattisia lausekkeita. Tehtäviä halutaan havainnollistaa myös graafisesti.

Järjestelmä ei saa kuormittaa palvelimia kohtuuttomasti. Jotta tehtävien tekeminen voidaan asettaa opintojakson suoritusvaatimukseksi, järjestelmän on toimittava luotettavasti useimmilla selaimilla, oltava käytettävissä mahdollisimman suuren osan ajasta ja kestettävä useiden kymmenien opiskelijoiden yhtäaikaista käyttöä. Järjestelmälle on oltava saatavilla teknistä tukea.

Opiskelijoiden hyväksyntään vaikuttavat heille näkyvät käyttöliittymät, tehtävien mielekkyys ja mahdollisuus vihjeisiin, malliratkaisuihin ja oppimateriaaliin. Opiskelijoiden on pystyttävä käyttämään järjestelmää ilman erillistä koulutusta, ja heidän huomionsa tulee keskittyä enemmän matematiikan tehtävän ratkaisemiseen kuin järjestelmän toiminnallisuuteen. Ohjelman tulee tukea ja ohjata opiskelijaa käyttämään hyviä, kuhunkin tehtävään soveltuvia ongelmanratkaisumenetelmiä.

Järjestelmän tulee mahdollistaa uusien tehtävien laatiminen. Tällä hetkellä käytössä oleva MathBridge-järjestelmä ei sisällä työkalua tehtävien luomiseen, minkä takia uusien tehtävien lisääminen järjestelmään vie aikaa ja vaatii teknistä osaamista. Tehtävien laatiminen halutaan siirtää järjestelmäylläpitäjiltä matematiikanopettajille.

Opiskelijoiden vastauksista kertyvää dataa halutaan kerätä moneen tarkoitukseen. Matematiikan opintojaksot koostuvat yleensä useammasta osasuorituksesta, esimerkiksi tentistä, laskuharjoituspaketista ja PC-harjoitus suorituksista. Järjestelmän on muodostettava opiskelijoiden järjestelmällä suorittamista tehtävistä pisteraportti, joka voidaan helposti yhdistää muihin opettajan ylläpitämiin pisteraportteihin. Opetuksen kehittämiseksi ja opetusta käsittelevän tieteellisen tutkimuksen mahdollistamiseksi myös opiskelijoiden vastaukset on tallennettava myöhempää tarkastelua varten. Järjestelmän tulee tallentaa myös muuta hyödyllistä dataa opiskelijoiden toiminnasta, kuten aikaleimoja ja virheellisiä yrityksiä.

Järjestelmästä voi aiheutua kuluja erilaisina lisenssi- ja aineistomaksuina sekä järjestelmäylläpitäjien ja opetushenkilökunnan palkkoina. Mitä enemmän järjestelmään käytetään rahaa, sitä selkeämpiä hyötyjä sen käytöstä on saatava.

Järjestelmän on integroiduttava muihin TTY:llä käytössä oleviin järjestelmiin. Vähimmäisvaatimus on mahdollisuus Shibboleth-kertakirjautumisjärjestelmän käyttöön. Integroitua Moodle-oppimisympäristöön toivotaan. Moodle on TTY:n virallinen oppimisympäristö, johon opintojaksot pyritään keskittämään. Vaikka itsenäinen ohjelma sisältää yleensä vähemmän riippuvuuksia, mikä voi tehdä ohjelman toiminnasta luotettavampaa, oppimisympäristöt tarjoavat enemmän toiminnallisuutta esimerkiksi opintojaksojen ja opiskelijoiden hallintaan, ja niiden laajentaminen on usein helpompaa. Myös oppimisympäristön tekninen ylläpito voidaan keskittää tehokkaasti. Opiskelijoidenkin on usein helpompaa käyttää yhtä oppimisympäristöä kuin tutustua uuteen järjestelmään joka opintojakson yhteydessä. Vielä suurempi merkitys on opettajien oikeuksilla ja mahdollisuuksilla hallita tehtäviä ja opintojaksoja. Yliopisto-opetuksessa ryhmäkoot voivat olla suuria: Samaa opintojaksoa voi suorittaa yhtäaikaaisesti useita satoja oppilaita. Opintojakson toteutukseen voi tällöin osallistua useita henkilöitä: Luennoitsijoita, muita opettajia ja erilaisia assistentteja. Yleensä heillä kaikilla on myös omat tehtävänsä ja oikeutensa opintojakson verkko-osissa. On myös tavallista, että sama henkilö osallistuu useamman opintojakson toteutukseen, jol-

loin on käytännöllistä, että kaikkia opintojaksoja voidaan hallinnoida saman järjestelmän kautta. Oppimisympäristöissä nämä ominaisuudet on yleensä sisäänrakennettu.

Uusien järjestelmien vertailukohteena pidetään nykyistä MathBridge-järjestelmää. Mahdollisen uuden järjestelmän toivotaan suoriutuvan sitä paremmin useimmilla vaatimusalueilla.

## 3 STACK

STACK on avoimen lähdekoodin työkalu interaktiivisten matematiikan tehtävien luomiseen ja käyttöön. Se valittiin tarkempaan tarkasteluun, koska sen käytöstä oli jo hie- man kokemuksia Tampereen teknillisellä yliopistolla, ja sitä on sekä tutkittu että käytet- ty enemmän Aalto-yliopistossa. Tämä luku esittelee STACKin eri versiot, rakenteen, tekniikat ja ominaisuudet.

### 3.1 Historia ja versiot

STACKista on olemassa kaksi virallista versiota, STACK1 ja STACK2. STACK3:n kehitystyö alkoi keväällä 2012. Lisäksi Aalto-yliopistolla on oma STACK- kehityshaaransa ja TTY:llä MathBridge-projektissa toteutettu STACKiin pohjautuva ActiveMath-integraatio (luku 2.5). Versioiden välillä on huomattavia eroja muun muas- sa ominaisuuksissa, käyttöliittymissä sekä oppimisympäristöintegraatioissa. Luku 3.1 esittelee ja vertailee eri kehityshaaroja.

#### 3.1.1 STACK1

STACK kehitettiin Iso-Britanniassa, University of Birminghamin matematiikan laitok- sella. STACKin kehitys alkoi vuonna 2004 ja ensimmäinen versio julkaistiin maalisi- kuussa 2005. Järjestelmän luoja Christopher Sangwin oli aiemmin käyttänyt opetukses- saan AiM-järjestelmää. STACKin ja AiMin suurimmat erot löytyvät tehtävien luomises- ta, vastausten tarkastamisesta ja käytetystä matematiikkaohjelmistosta.

STACKin suunnittelun kaksi tärkeintä tavoitetta olivat opiskelijoiden syöttämien lau- sekkeiden algebrallinen tarkastus ja opettajien tehtävien luominen. Toisin kuin AiM, STACK ei oleta matematiikanopettajan osaavan tietokoneohjelmointia. STACKin luojat uskoivat, että automaattisesti tarkastettavien matematiikan tehtävien käyttöönottoa hi- dasti erityisesti matematiikanopettajien työkalujen jyrkkä oppimiskäyrä. Heidän mieles- tään järjestelmän käyttöönoton tulee olla nopeaa ja opettajien on voitava mukauttaa sitä omaan opetukseensa soveltuvaksi. (Sangwin & Grove 2006). STACK mahdollistaa opiskelijan vastauksen tarkastamisen puurakenteella, jonka opettaja voi luoda opettajan käyttöliittymässä ilman ohjelmointikomentoja. Opettajan on kuitenkin tunnettava aina- kin joitakin Maxima-, LaTeX- ja HTML-komentoja.

Sekä STACK että AiM ovat ilmaisia, avoimen lähdekoodin järjestelmiä. STACKin ma- tematiikkaohjelmistoksi on kuitenkin valittu ilmainen, avoimen lähdekoodin Maxima,

kun AiM käyttää kaupallista Maplea, johon käyttäjäorganisaation on ostettava lisenssi (Sangwin 2003).

STACK on itsenäisesti toimiva järjestelmä, joka sisältää kaksi erillistä näkymää: opiskelijan ja opettajan käyttöliittymät. Opiskelijat voivat luoda itselleen käyttäjätunnuksia ja vastata tehtäviin. Opettajat voivat luoda kursseja, tehtäväsarjoja ja yksittäisiä tehtäviä, ottaa opiskelijoita kursseille ja katsella raportteja opiskelijoiden tekemistä tehtävistä.

### 3.1.2 OtaSTACK

Aalto-yliopisto<sup>2</sup> otti STACKin käyttöön syksyllä 2006. Se on STACKin suurin yksittäinen käyttäjä: STACK-tehtäviä käytetään edelleen kursseilla, joille osallistuu vuosittain useita tuhansia opiskelijoita (Sangwin 2010).

Aalto-yliopisto on tehnyt STACK1:een useita muutoksia. Tätä paranneltua versiota kutsutaan Aalto-yliopiston teknillisten alojen korkeakoulujen kotipaikan Otaniemen mukaan OtaSTACKiksi erotuksena alkuperäiseen STACK1:een. Eräät lähteet käyttävät myös nimeä modSTACK (Ruokokoski 2009). OtaSTACKin tärkeimmät erot STACK1:een ovat seuraavat:

- STACK1:n käyttäjätunnistus on korvattu Aalto-yliopiston käyttämällä kertakirjautumisjärjestelmällä.
- STACK1 esittää tehtävänannot LaTeX-ladontajärjestelmän ja TtH-ohjelman avulla, mikä voi aiheuttaa ongelmia joissakin selaimissa. OtaSTACK esittää kaikki tehtävänannot kuvamuodossa selainyhteensopivuusongelmien välttämiseksi.
- OtaSTACK kerää kattavampia lokitietoja järjestelmän käytöstä.
- OtaSTACKin istunnonhallinta mahdollistaa useamman STACK-järjestelmän samanaikaisen käyttämisen.
- Sekä STACK1:ssä että OtaSTACKissa opettaja voi halutessaan määrittää, että väärät vastausyritykset vähentävät opiskelijan tehtävästä saamaa pistemäärää. STACK1:ssä jokainen väärä vastausyritys vähentää pisteitä automaattisesti. OtaSTACKissa opettaja voi antaa opiskelijalle kolme ilmaista vastausyritystä, jolloin tehtävästä saatavat pisteet alkavat vähentyä vasta neljännellä väärällä vastauksella.
- OtaSTACK sisältää ehto- ja silmukkarakenteet (if ja for), jotka mahdollistavat muun muassa graafien piirtämisen.
- OtaSTACK tukee avoimen lähdekoodin Graphviz-ohjelmaa, jolla voidaan visualisoida graafeja.

(Harjula 2008, Ruokokoski 2009)

---

<sup>2</sup> Aalto-yliopisto aloitti toimintansa 1.1.2010 Teknillisen korkeakoulun, Kauppakorkeakoulun ja Taideteollisen korkeakoulun yhdistyessä. Tätä ennen STACKin parissa työskenteli Teknillisen korkeakoulun Matematiikan laitos, joka on nykyään osa Aalto-yliopiston Perustieteiden korkeakoulua.

### 3.1.3 STACK2

STACK2:n tärkein ero edeltäjänsä on opiskelijan käyttöliittymän poistuminen. Opettaja voi edelleen luoda kysymyksiä opettajan käyttöliittymässä, jota tosin on uudistettu huomattavasti, mutta tehtäviä ei ole mahdollista ratkaista, ellei STACKia integroida erilliseen oppimisympäristöön. Moodle 1.9 on ainoa tuettu vaihtoehto, mutta integraatio muihin ohjelmistoihin on periaatteessa mahdollista.

Moodle-integraatio on osoittautunut hyväksi ratkaisuksi, sillä opiskelijoiden ja opinto-suoritusten hallinta eivät olleet STACK1:n parhaita puolia. Integraatio erilliseen oppimisympäristöön mahdollistaa STACK-kehittäjien keskittymisen juuri matematiikan tehtäviin liittyviin kysymyksiin, kun oppimisympäristön kehittäjät vastaavat hallinnollisista ja muuhun opetukseen liittyvistä asioista. Työnjakoa voidaan tehdä myös opettajien kesken: Matematiikanopettaja voi luoda tehtävät STACKin avulla ja opintojakson pää-assistentti voi huolehtia Moodlessa opiskelijoiden ilmoittautumisista, suoritusmerkintöjen kirjaamisesta ja muista opintojaksoon liittyvistä käytännön järjestelyistä. STACKin ja Moodlen integroiminen tekee molemmista monipuolisempia ja houkuttelevampia vaihtoehtoja opetukseen.

Moodle-integraatio on toteutettu Opaque-kysymystyyppin avulla. Opaque on SOAP-pohjainen protokolla, joka mahdollistaa Moodlen tentti-työkalun keskustelun erillisen kysymysmoottorin kanssa. (Hunt 2011)

Muita STACK2:ssa esiteltyjä uusia ominaisuuksia ovat muun muassa matematiikan esitys JSMath-JavaScript-kirjaston avulla, moniosaiset kysymykset, välimuisti, ehtolausekkeet sekä joukko uusia vastauksia ja interaktiivisia elementtejä (StackWiki 2011b). Näistä moniosaiset kysymykset ja ehtolausekkeet oli toteutettu jo OtaSTACKissa.

STACK2:n viimeisin julkaistu versio on 2.2. Lisäksi saatavilla on kehitysversio. Mahdollisista tulevista versioista ei ole tiedotettu, ja on todennäköistä, ettei STACK2:n kehitys enää jatku kehittäjien keskittyessä STACK3:en. STACK2 on käytössä ainakin kymmenessä oppilaitoksessa (Sangwin 2010).

### 3.1.4 STACK3

STACKin pääasiallinen kehittäjä Christopher Sangwin ja Moodlen Opaque-protokollan luoja Tim Hunt työskentelivät keväällä 2012 STACKin kolmannen version parissa. Kuten STACK2, myös STACK3 tulee toimimaan osana oppimisympäristö Moodlea. STACK3 tulee kuitenkin toimimaan Moodlen versioissa 2.x (StackWiki 2011b).

STACK2:n ja STACK3:n suurin ero on Moodle-integraation laatu. STACK2 on erillinen ohjelma, joka yhdistetään Moodleen Opaque-kysymystyyppin avulla. STACK3 on

Moodlen natiivi kysymystyyppi, jolla ei ole erillistä käyttöliittymää (Sangwin 2012). Kysymystyyppi on Moodlen sisäinen malli, johon kuuluu sekä tietosisältöä että toiminnallisuutta. Kysymystyyppi määrittelee, kuinka tehtävänanto ja vastauksen palautus ja mahdollinen koneellinen tarkastus suoritetaan. Sangwinin mukaan STACK3:n toteutus Moodle-kysymystyyppinä helpottaa kehitystyöhön osallistumista ja STACKin asennusta, vähentää riippuvuuksia ja tarvittavien erillisten komponenttien määrää ja parantaa STACK-tehtävien ja Moodlen tentti-työkalun yhteistoimintaa (Sangwin 2012).

STACK3:sta ei kuitenkaan ole vielä julkaistu edes kehitysversiona, joten sitä ei käsitellä tässä diplomityössä.

### 3.1.5 Vertailu

STACK-version valintaan vaikuttaa useita tekijöitä. Tampereen teknillisellä yliopistolla valintaprosessista karsiutuivat ensimmäisinä STACK1 ja STACK3. Vaikka STACK3 vaikuttaa lupaavalta, sen kehitys on vielä alkuvaiheissa, eikä käyttöönotto ole mahdollista halutussa aikataulussa. STACK1 on stabiili ohjelma, jonka käytöstä on muissa yliopistoissa saatu jo tuloksia. Siitä on kuitenkin olemassa paranneltu versio, joka sisältää kaiken STACK1:n toiminnallisuuden sekä eräitä hyödyllisiä lisäyksiä. Vaihtoehtoiksi jäivät siis OtaSTACK ja STACK2, joiden ominaisuuksia vertaillaan taulukossa 3.1.

Kolmanneksi vertailukohteeksi valittiin MathBridge (luku 2.5), johon Tampereen teknillinen yliopisto on itse toteuttanut STACK-tyylisen tehtävämoottorin. MathBridgen mukaan vertailuun ottamisen tarkoitus on arvioida, onko tarkoituksenmukaisempaa käyttää jotakin valmiista STACK-versioista vai käyttää niitä apuna oman työkalun luomiseen.

Järjestelmiä vertailtiin luvussa 2.6 esiteltyjen seitsemän vaatimuksen avulla. Tulokset on esitetty taulukossa 3.1.

OtaSTACKin ja STACK2:n tehtävien tason todettiin riittävän yliopistotasoiseen matematiikanopetukseen. MathBridgen ominaisuuksista puuttuvat kuvien esitys ja tarkastuspuut.

Joka järjestelmällä on oma tapansa esittää matemaattisia merkintöjä. OtaSTACK käyttää ensin TtH-ohjelmaa muuttaakseen tehtävänantojen ja mallivastausten LaTeX-merkinnät HTML-koodiksi. TtH:n tuottama HTML voidaan kuitenkin esittää eri selaimissa eri tavoin. Siksi OtaSTACK muuttaa tehtävänannot ja mallivastaukset palvelinpäässä PNG-kuviksi, jotka upotetaan sivuille. Näin varmistetaan, että kaikki opiskelijat näkevät tehtävänannot sellaisina kuin ne on tarkoitettu. Kuvat tallennetaan erilliseen tietokantaan, jolloin niitä ei tarvitse luoda uudestaan jokaisen sivulatauksen yhteydessä. Tämän menetelmän käyttö vaatii kuitenkin hieman lisätyötä ohjelman asennusvaiheessa. STACK2 ja MathBridge käyttävät LaTeX-merkintöjen esittämiseen erillisiä JavaSc-



ript-kirjastoja. STACK2 käyttää oletuksena jsMathia, MathBridge MathJaxia. JsMathin kehitys on jo lopetettu, minkä takia sen seuraajan MathJaxin on todettu toimivan luotettavammin nykyisin käytössä olevissa selaimissa. Tampereen teknillisellä yliopistolla se onkin integroitu sekä OtaSTACKiin että STACK2:een. Matemaattisten merkintöjen oletusesityksellä ei siis ole enää merkitystä järjestelmän valinnassa.

**Taulukko 3.1.** OtaSTACKin, STACK2:n ja MathBridgen tehtävämoottorin eroja.

	<b>OtaSTACK</b>	<b>STACK2</b>	<b>MathBridgen tehtävämoottori</b>
<b>Tehtävien taso</b>	Riittävä	Riittävä	Välttävä
<b>Matemaattiset merkinnät</b>	On	On	On
<b>Suorituskyky</b>	Riittävä	Välttävä	Riittävä
<b>Käyttäjät</b>	Aalto-yliopisto	Muutama oppilaitos, myös Aalto-yliopisto <sup>3</sup>	MathBridge-ohjelmaan osallistuvat yliopistot ja lukiot ympäri Eurooppaa
<b>Tuki</b>	Mahdollisesti Aalto-yliopistosta	Mahdollisesti kehitysyhteisön kautta	Mahdollisesti DFKI:lta
<b>Tulevaisuuden näkymät</b>	Kehitys loppunut, Aalto ylläpitää toistaiseksi	Kehitys loppunut	TTY ylläpitää toistaiseksi. DFKI jatkokehittää ActiveMath-alustaa.
<b>Opiskelijoiden mielipide</b>	Positiivinen?	Ristiriitainen	Ristiriitainen
<b>Mahdollisuus omien tehtävien laatimiseen</b>	On	On	Ei
<b>Vastausraportit</b>	Saatavilla	Saatavilla	Puutteelliset
<b>Käyttökustannukset</b>	?	?	Pienet
<b>Stand-alone</b>	On	Vain opettajan ympäristö	Ei ole
<b>Integraatio oppimisalustoihin</b>	Ei mahdollista	Moodle	ActiveMath

MathBridge on keveytensä ansiosta huomattavasti OtaSTACKia ja STACK2:ta nopeampi. Kummatkin STACKit ovat kuitenkin Maximian optimoinnin jälkeen riittävän no-

<sup>3</sup> (Sangwin 2010)

peita opetuskäyttöön. STACK2 käyttää tietokantaa välimuistina ja tarvitsee siksi paljon tietokantatilaa.

Käyttäjien ja saatavilla olevan tuen määrä antaa vihjeitä siitä, kuinka paljon työtä järjestelmän ylläpito vaatii. OtaSTACK on stabiili ja tukeakin on mahdollista saada, mutta se lepää kokonaan Aalto-yliopiston varassa. Ylläpito loppuu, kun Aalto siirtyy uuteen järjestelmään. STACK2:lla on käyttäjiä, muttei enää kehitystä kehittäjiensä siirtyessä STACK3:een. MathBridgen jatko ei keväällä 2012 ole vielä selvillä. Järjestelmä on kuitenkin stabiili. Koska kyseessä on TTY:n itse kehittämä ja ylläpitämä tehtävämoottori, sen ylläpitoa voidaan jatkaa niin kauan kuin on tarve.

Opiskelijoiden hyväksynnästä voidaan esittää vasta arvauksia. Aalto-yliopiston opiskelijoiden kokemukset OtaSTACKista ovat positiivisia. Opiskelijan mielipiteet STACK2:sta (luku 4.2) ja MathBridgesta (Embacher et al. 2011) jakautuivat. STACK2:ssa opiskelijat kokivat ongelmia käyttöliittymän ja syntaksin kanssa, MathBridgessä järjestelmän teknisen toimivuuden.

OtaSTACKissa ja STACK2:ssa opettajan on mahdollista luoda uusia tehtäviä. MathBridgessä uusien tehtävien luominen on mahdollista vain järjestelmän ylläpitäjille.

OtaSTACKin ja STACK2:n raportointiominaisuudet on havaittu riittäviksi. STACK2:ssa erityisen hyvää on se, että muiden opintojaksoon kuuluvien Moodletehtävien, kuten monivalintakyselyjen tai esseiden, arvosanat saadaan samaan raporttiin. MathBridgen ActiveMath-alustan tarjoamat raportit ovat melko vaikeita käyttää ja joissain kohdin jopa selvästi virheellisiä. Tampereen teknillisellä yliopistolla niiden tarjoama data on jouduttu erikseen jalostamaan hyödyllisempään muotoon.

OtaSTACKin ja STACK2:n käyttökustannuksista voidaan esittää vain arvioita. Vaikka järjestelmät ovat ilmaisia, niiden käyttö vaatii huomattavasti henkilöresursseja sekä teknisen että opetushenkilökunnan puolelta. Kumman tahansa STACKin käyttöönotto matematiikan perusopintojaksoilla vaatii muutaman henkilötyövuoden. MathBridge on valmiina järjestelmänä edullisempi ylläpidettävä.

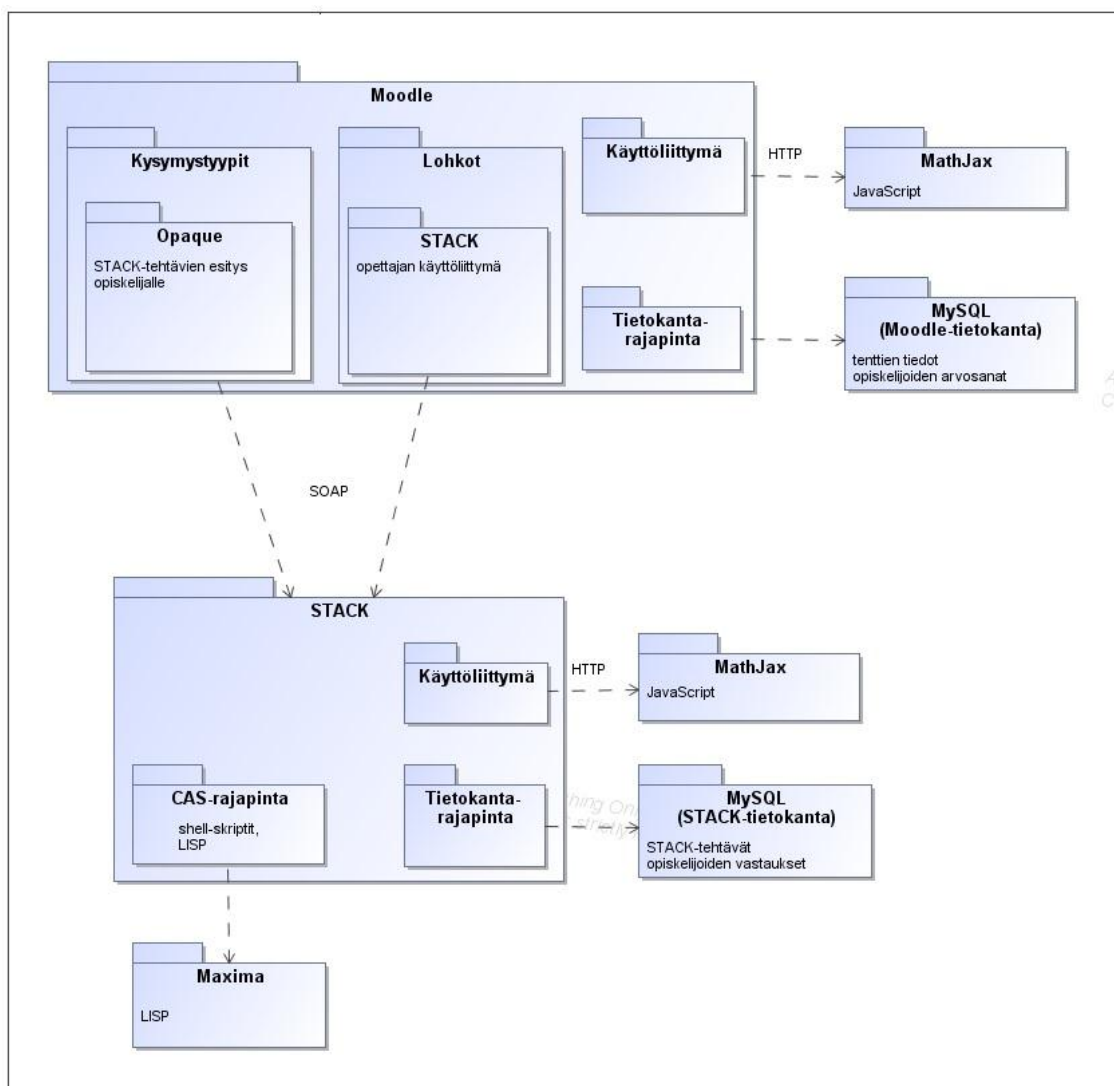
OtaSTACK ei sisällä STACK1:n istuntojen ja opiskelijoiden hallintaan liittyviä ominaisuuksia, vaan olettaa järjestelmän käyttäjän toteuttavan ne itse. STACK2 käyttää oletuksena Moodlen opiskelijanhallintaa, MathBridge ActiveMathin. Tampereen teknillinen yliopisto on integroinut Shibboleth-kertakirjautumisjärjestelmän OtaSTACKiin, Moodleen ja ActiveMathiin. Kertakirjautumisjärjestelmän käytön takia järjestelmän alkupe-  
räisellä istunnon- ja käyttäjänhallinnalla ei ole suurta merkitystä järjestelmän valinnassa.

STACK2:n käyttämä Moodle on integraationa ylivoimainen, vaikka itse STACKtehtävien katseluun ja muokkaukseen liittyvissä oikeuksissa olisikin vielä hiottavaa.

OtaSTACKin pullonkaula on ylläpitäjän käyttäjätunnus: Sillä ei ole järkevää luoda kuin yksi yhden opettajan ylläpitämä opintojakso. MathBridge ei sisällä minkäänlaista opettajan käyttöliittymää tehtävien luomiseen. Tehtävät on luotava muilla keinoilla ja tuotava järjestelmään XML-dokumentteina.

### 3.2 Tekninen kuvaus

Tämä luku esittelee tarkemmin STACK2:n rakennetta ja toimintaa. Tarkasteltavaksi järjestelmäksi valittiin STACK2, koska sen Moodle-integraatio antaa näkökulman siihen, kuinka matematiikan tehtäviä on mahdollista sulauttaa osaksi muuta verkko-opetusta. Kuva 3.1 esittää STACKin keskeiset komponentit.



**Kuva 3.1.** STACKin komponentit.

Rasila, Harjula ja Zenger totesivat matematiikan tehtäviä automaattisesti tarkastavan järjestelmän koostuvan seuraavista osista:

1. käytönvalvonta ja käyttäjänhallinta opiskelijoiden ja opettajien tunnistamiseen

2. opiskelijan käyttöliittymä (tehtävien ratkaisemiseen)
3. opettajan käyttöliittymä (uusien tehtävien luomiseen)
4. tietokanta, johon tehtävät tallennetaan
5. matematiikkaohjelmisto, jota käytetään vastausten arvosteluun ja palautteen antamiseen, ja
6. arvosanataulukko, johon opiskelijoiden suoritukset tallennetaan.

(Rasila et al. 2007)

Edellisessä jaossa järjestelmän oletetaan toimivan itsenäisesti. STACK2:n Moodle-integraatio tekee kohdista (1) ja (6) STACKin kannalta triviaaleja, sillä Moodle huolehtii niiden toteuttamisesta. Se saa myös aikaan tarpeen työkalulle, joka yhdistää kaksi järjestelmää toisiinsa. Seuraavat aliluvut käsittelevät listan kohtia (2)-(5) sekä Opaque-protokollaa.

### 3.2.1 Opiskelijan käyttöliittymä

Toisin kuin STACK1 ja OtaSTACK, STACK2:een ei kuulu erillistä opiskelijan käyttöliittymää. Opiskelijat käyttävät STACKia vain Moodlen käyttöliittymän kautta, eikä heidän pitäisi edes huomata käyttävänsä erillistä järjestelmää. Moodle on internetissä toimiva ohjelma, jota voidaan käyttää moderneilla selaimilla, jotka tukevat JavaScriptiä ja evästeitä.

JavaScriptiä käytetään paitsi Moodlen omiin toimintoihin, myös matemaattisten merkintöjen esittämiseen selaimessa. Tampereen teknillisellä yliopistolla tarkoitukseen käytetään ilmaista avoimen lähdekoodin MathJax-kirjastoa. MathJax muuttaa LaTeX-ladontakielellä kirjoitetun tekstin käyttäjän valinnan mukaan joko HTML:ksi ja CSS:ksi, MathML:ksi tai SVG-kuvaksi. Lähes kaikki selaimet pystyvät esittämään ainakin yhtä näistä muodoista.

STACK-tehtävät esitetään opiskelijalle ”tentteinä”. Tenti on Moodlen nimitys arvioitavalle tehtäväsarjalle, joka voi sisältää muitakin kuin STACK-tehtäviä. Opiskelija valitsee opintojakson Moodle-sivulta haluamansa tentin, jonka jälkeen hän saa siitä lisää informaatiota, kuten tentin aukioloajan tai jäljellä olevien yrityskertojen määrän, ja voi valita, haluaako aloittaa tentin suorittamisen.

Kuva 3.2 esittää erästä STACK-tehtävää sellaisena, kun opiskelijat sen näkevät. STACK-tehtävät esitetään opiskelijalle yksi kerrallaan. Opiskelija voi kuitenkin katsella ja ratkaista STACK-tehtäviä haluamassaan järjestyksessä Sivu-linkkien kautta. Syötettyään vastauksensa opiskelijan on ensin tarkastettava vastauksen syntaksin oikeellisuus painamalla Submit-painiketta. Jos syntaksissa ei ole virheitä, opiskelija voi seuraavaksi tarkastaa vastauksen oikeellisuuden painamalla uudestaan Submit-painiketta. Kuva 3.3 näyttää tarkastuksen tuloksen oikealla ja väärällä vastauksella.

**STACK-tehtävä** Olet kirjautunut nimellä Teemu Teekkari. (Kirjaudu ulos)

Moodle 1.9 > STACK-101 > Tentit > Kotitehtävä 2 > Yritys 1

### Kotitehtävä 2 - Yritys 1

Sivu: 1 2 (Seuraava)

**1**  
Pistettä: 0/2

Laske käyrän  $\vec{r}(t) = (\cos(2 * t), \sin(3 * t))$  derivaattavektori.

$\vec{r}'(t) =$

Syötä vastauksesi muodossa [lauseke1, lauseke2]

Laske derivaatan arvo, kun  $t = 1$ .

$\vec{r}'(1) =$

Syötä vastauksesi muodossa [arvo1, arvo2]

Sivu: 1 2 (Seuraava)

**Kuva 3.2.** STACK-tehtävän tehtävänannon esitys opiskelijalle.

Jos vastaus ei ole kokonaan oikea, opiskelija voi halutessaan yrittää uutta vastausta. Opiskelija voi koska tahansa siirtyä ratkaisemaan toista tehtävää tai jatkaa keskeneräisen tehtävän ratkaisua. Opiskelija voi myös koska tahansa lopettaa tentin painamalla Palauta kaikki ja lopeta -painiketta. Kun opiskelija lopettaa tentin, hänelle näytetään yhteenveto vastauksista ja tentistä saatu kokonaispistemäärä. Opiskelija voi halutessaan tarkastella vastauksia ja pisteitään koska tahansa valitsemalla opintojakson Moodle-sivulta haluamansa tentin ja haluamansa yrityskerran.

Opiskelijan käyttöliittymässä haastavinta on hahmottaa, mitkä painikkeet liittyvät STACKiin ja mitkä Moodleen, ja mitä tulee käyttää missäkin tilanteessa. Esimerkiksi tentin lopettaminen onnistuu sekä STACKin Finish quiz -painikkeesta että Moodlen Palauta kaikki ja lopeta -painikkeesta. Opiskelijan on myös muistettava, että Submit-painiketta on aina käytettävä vähintään kahdesti, ensin vastauksen syntaksin validointiin ja sen jälkeen varsinaiseen tarkastamiseen. Jos opiskelija unohtaa jälkimmäisen painalluksen, vastausta ei arvostella eikä pisteitetä. Se tallentuu kuitenkin jo validoinnin yhteydessä, joten opettajan on mahdollista arvostella se myöhemmin manuaalisesti.

**STACK-tehtävä** Olet kirjautunut nimellä Teemu Teekkari. (Kirjaudu ulos)

Moodle 1.9 ▶ STACK-101 ▶ Tentit ▶ Kotitehtävä 2 ▶ Yritys 2

### Kotitehtävä 2 - Yritys 2

Sivu: 1 2 (Seuraava)

**1**  
Pistettä: 1/2 Laske käyrän  $\vec{r}(t) = (\cos(4 * t), \sin(5 * t))$  derivaattavektori.

$\vec{r}'(t) = [-4*\sin(4*t), 5*\cos(5*t)]$

Syötä vastauksesi muodossa [lauseke1, lauseke2]

Your last answer was interpreted as:  
 $[(-4) \cdot \sin(4 \cdot t), 5 \cdot \cos(5 \cdot t)]$

**Correct answer, well done.**  
Your mark for this attempt is 1. ✔ With penalties, and previous attempts, this gives 1 out of 1

Laske derivaatan arvo, kun  $t = 1$ .

$\vec{r}'(1) = [-4, 5]$

Syötä vastauksesi muodossa [arvo1, arvo2]

Your last answer was interpreted as:  
 $[-4, 5]$

**Incorrect answer.**  
The entries in red below are those that are incorrect.  
 $[-4, 5]$

Your mark for this attempt is 0. ✘ With penalties, and previous attempts, this gives 0 out of 1

**Kuva 3.3.** STACK-tehtävän arvioinnin esitys opiskelijalle.

### 3.2.2 Tehtävän luominen

Tehtäviä voidaan luoda STACK2:ssa kahdesta eri käyttöliittymästä: Moodlen kautta opettajatunnuksella tai erillisen ylläpitäjän käyttöliittymän kautta ylläpitäjän tunnuksetta. Käyttöliittymät ovat lähes identtiset. Suurimmat erot ovat kysymysten lisääminen Moodlen kysymyspankkeihin, joka toimii vain Moodlen kautta, ja ylläpitäjän käyttöliittymästä löytyvä testisarja, jolla voidaan etsiä järjestelmän toimintaan liittyviä ongelmia. Tässä luvussa kuvatut toiminnot ovat yhteisiä molemmille käyttöliittymille. Opettajan näkemä tehtäväpankki esitetään kuvassa 3.4.

**Moodle 1.9** ► Questions available from STACK2 question engine

[New Question](#) | [Import a Question](#) | [CasChat](#) | [Question Search](#) | [Reports](#) | [Stack Documentation](#)

keywords: \*, arvaus, itseisarvo, kuvaaja, logaritmi, matriisi, murtoluku, neliöjuuri, polynomifunktio.

Most common keywords: murtoluku, neliöjuuri, polynomifunktio, matriisi, logaritmi, itseisarvo, kuvaaja, arvaus.

6 question(s)    case sensitive

<input type="checkbox"/> Name: Description keywords	Valid	#	Privacy	Date Edited	Categories	Actions
<input type="checkbox"/> <b>Osittaisderivaatta:</b>	✓	15	*	2:13pm	Oletus kohteelle Stackin testauskurssi	<a href="#">?</a> <a href="#">✖</a> <a href="#">»</a> <a href="#">📄</a>
<input type="checkbox"/> <b>Kuvaajan tunnistus:</b> polynomifunktio, kuvaaja kuvaaja, polynomifunktio	✓	0	*	11:27pm, 10 Apr	Oletus kohteelle Stackin testauskurssi	<a href="#">?</a> <a href="#">✖</a> <a href="#">»</a> <a href="#">📄</a>
<input type="checkbox"/> <b>Arvaa luku:</b> arvaus arvaus	✓	10	*	11:27pm, 10 Apr	Oletus kohteelle Stackin testauskurssi	<a href="#">?</a> <a href="#">✖</a> <a href="#">»</a> <a href="#">📄</a>
<input type="checkbox"/> <b>Taikaneliö:</b> matriisi matriisi	✓	1	*	11:27pm, 10 Apr	Oletus kohteelle Stackin testauskurssi	<a href="#">?</a> <a href="#">✖</a> <a href="#">»</a> <a href="#">📄</a>
<input type="checkbox"/> <b>Sievennys:</b> murtoluku, itseisarvo, logaritmi, neliöjuuri itseisarvo, logaritmi, murtoluku, neliöjuuri	✓	1	*	11:27pm, 10 Apr	Oletus kohteelle Stackin testauskurssi	<a href="#">?</a> <a href="#">✖</a> <a href="#">»</a> <a href="#">📄</a>

[invert selection](#)

**With selected**

**Kuva 3.4.** STACKin tehtäväpankki.

Uuden tehtävän luomisen käyttöliittymä koostuu seitsemästä osasta, jotka esitetään kuvissa 3.5-3.11. Kuvassa 3.5 näkyvät myös käyttöliittymän tallennustoiminnot. Seuraavaksi esitetään lyhyesti, mitä tietoja tehtävän luomiseen tarvitaan. Tarkemmat ohjeet löytyvät opettajan ohjeista.

Aluksi (kuva 3.5) tehtävälle annetaan nimi (name). Description- ja keywords-kenttien sisällöt näkyvät tehtäväpankissa ja antavat lisätietoa tehtävästä. Question variables -kentässä määritellään CASText-tyyppisissä kentissä käytettävät muuttujat, joita ei johdeta opiskelijan vastauksesta. Kentässä voidaan käyttää Maxima-funktioita. Question stem -kenttään syötetään tehtävänanto. Siinä voidaan käyttää HTML:ää, LaTeXia ja question variables -kohdassa määriteltyjä muuttujia, yhteisnimitykseltään CASTextiä, sekä seuraavia erikoismuuttujia:

- #muuttuja# opiskelijan vastaukselle
- <IEfeedback>muuttuja</IEfeedback> interaction elements -kohdan tuottamalle palautteelle ja
- <PRTfeedback>vastauspuun numero</PRTfeedback> potential response trees -kohdan tuottamalle palautteelle.

Worked solution -kenttään opettaja voi halutessaan syöttää tehtävän malliratkaisun. Question note -kenttään kirjoitetaan tunniste, joka yksilöi saman tehtävän eri instanssit. Jos tehtävässä käytetään satunnaistettuja parametreja, ne kirjataan tähän. Sekä worked solution että question note voivat sisältää CASTextiä.

Moodle 1.9 ► Stack Questions ► Author a question on STACK2

Update

Save

Save as New

Export as XML

- Stem
- Interactions
- Responses
- Options
- Tests
- Meta

**Remember:** Finish  
Authoring with button at  
bottom of page.

Name:

Description:

Keywords:

Question Variables:

**Question Stem:**

#ans# denotes student answers  
 @castext@ for castext  
 <html></html> for html  
 latex

Preview

#ans1# <IEfeedback>ans1</IEfeedback>

Worked Solution:

Preview

Question Note:

**Kuva 3.5.** Tehtävän luomisen käyttöliittymän ensimmäinen osa.

Interactions elements -osassa (kuva 3.6) jokaista opiskelijan vastausta varten luodaan validointitoiminto. Sen tärkeimmät osat ovat input type, joka määrittää, tarkastellaanko vastausta esimerkiksi lausekkeena, matriisina tai monivalintana. Teacher's answer -kenttään syötetään kyseisen tehtävänosan oikea vastaus. Tavallisesti oikea vastaus määritellään question variables -kohdassa ja interaction elementsiin kirjoitetaan vain kyseisen muuttujan nimi. Muut interaction elementsien osat riippuvat valituista syötemuodoista. Niissä tarkastellaan ensisijaisesti vastauksen oikeamuotoisuutta, ei varsinaista oikeellisuutta. Box size määrittää opiskelijalle näytettävän vastauskentän leveyden. Strict syntax ja insert stars lisäävät opiskelijan vastaukseen puuttuvia kertomerkkejä. Oletusarvoilla strict syntax true ja insert stars false kertomerkkejä ei lisätä. Syntax hint –




kenttään opettaja voi syöttää syntaksivihjeen, esimerkiksi vektorin syntaksin  $[?, ?, ?]$ . Forbidden words määrittää Maxima-funktiot, joita opiskelija ei saa käyttää vastauksessaan. Esimerkiksi  $\text{diff}()$  on syytä kieltää derivointitehtävissä. Vaihtoehtoisesti allowed words -kentällä voidaan määrittää syötteessä sallitut funktiot, jolloin muut kielletään. Forbid floats määrittää, voiko opiskelija antaa vastauksena desimaaliluvun. Oletusarvoisesti desimaaliluvut on kielletty. Require lowest terms hyväksyy vastaukseksi vain supistetun muodon. Check students answer's type tarkastaa, että opiskelijan vastaus on samaa muotoa kuin opettajan. Jos opettajan vastaus on esimerkiksi vektori, myös opiskelijan on syötettävä vektori. Student must verify määrittää, suoritetaanko validointi ja arvostelu kahtena erillisenä vaiheena. Hide feedback piilottaa opettajan palautteen opiskelijalta. Input type options -valintoja käytetään, kun vastauksen tyyppi on monivalinta. Monivalintatehtävien luomista STACKilla ei suositella.

▼ Interaction Elements	
	Students's Answer Key <b>ans1</b>
Input Type	Algebraic Input
Teacher's Answer	<input type="text"/> This is a required field.
Box Size	20
Strict Syntax	True
Insert Stars	False
Syntax Hint	<input type="text"/>
Forbidden words	<input type="text"/>
Allowed words	<input type="text"/>
Forbid Floats	True
Require lowest terms	True
Check Students answer's type	Check types
Student must verify	True
Hide feedback	False
Input Type Options	No options

**Kuva 3.6.** Tehtävän luomisen käyttöliittymän interaction elements -osa.

Potential response trees -osassa (kuva 3.7) määritellään, kuinka opiskelijan vastaus tarkastetaan. Tarkastuspuille voidaan määritellä omia muuttujia (feedback variables), joissa voidaan Maxima-funktioiden lisäksi käyttää myös opiskelijan vastauksia. Näin voidaan tarkastaa esimerkiksi onko opiskelijan vastaus derivoituva. Tehtävään liittyvien tarkastuspuiden määrä ei riipu tehtävän vastauskohtien määrästä, ja samaan lopputulokseen on usein mahdollista päästä useilla tarkastuspuuyhdistelmillä. Jokainen tarkastuspuu suorittaa jonkin tarkastusoperaation, ja opiskelijan lopullinen pistemäärä on kaikkien tarkastuspuiden pisteiden summa. Yksittäinen tarkastuspuu saa arvon 0-1, joka kerrotaan question value -kentän kertoimella.


**▼ Potential Response Trees** 

Add a potential response tree named:

---

**1**

Question Value:  Auto Simplify:

Feedback Variables: 

Description:

Add  Potential Responses:

**No: 0**

SAns:  TAns:  Answer test:

Quiet:

**Teachers Answer: This is a required field.**  
**Students Answer: This is a required field.**

true	false
Mod: <input type="text" value="="/> Mark: <input type="text" value="1"/> Penalty: <input type="text"/> Next PR: <input type="text" value="-1"/>	Mod: <input type="text" value="="/> Mark: <input type="text" value="0"/> Penalty: <input type="text"/> Next PR: <input type="text" value="-1"/>
Feedback: <input type="text"/>	Feedback: <input type="text"/>
Answer Note: <input type="text" value="1-0-T"/>	Answer Note: <input type="text" value="-0-F"/>
Teachers Notes: <input type="text"/>	
<input data-bbox="1220 1590 1340 1624" type="button" value="Remove"/>	

**Kuva 3.7.** Tehtävän luomisen käyttöliittymän potential response trees -osa.

Tarkastuspuu koostuu solmuista (potential response), joista jokainen vertaa yhtä opiskelijan vastausta (SAns) yhteen oikeaan vastaukseen (TAns). Answer test -kenttä määrittää samankaltaisuuden kriteerit käyttäen tarvittaessa Test ops -kenttään syötettäviä parametreja. Jokainen solmu voi tarkastuksen perusteella muuttaa tarkastuspuun arvosanaa Mod-, Mark- ja Penalty-kenttien määrittämällä tavalla, näyttää opiskelijalle Feedback-kenttään syötetyn palautteen ja siirtää vastauksen tarkastelun toiseen, Next PR -kentässä

määritettyyn solmuun. Answer note -kenttää käytetään tallentamaan tieto siitä, mistä solmusta arvosana tai palaute on lähtöisin. Teachers Notes -kenttään opettaja voi kirjata omia muistiinpanojaan, joita ei näytetä opiskelijoille.

Options-osassa (kuva 3.8) tehtäville voidaan määrittää erilaisia asetuksia. Tehtävän luomisen muista osista poiketen näitä asetuksia on mahdollista kopioida tehtävästä toiseen erillisen käyttöliittymän kautta. Question level simplify asettaa Maximian simp-muuttujan, joka määrittää, sievennetäänkö lausekkeet yksinkertaisimpaan muotoonsa. Assume positive asettaa Maximian assume\_pos-muuttujan, joka määrittää, pidetäänkö etumerkitöntä muuttujaa automaattisesti positiivisena. Question penalty määrittää, kuinka monta prosenttia maksimipisteistä vähennetään jokaisesta väärästä vastauksesta. Mark modification määrittää, onko pisteiden vähennys käytössä. Feedback used määrittää, näytetäänkö opiskelijalle palautetta kaikista, oikeista, osittain oikeista vai vääristä vastauksista. Feedback correct, Feedback partially correct ja Feedback incorrect -kenttiin opettaja voi syöttää haluamansa palautetekstin. Worked solution on demand -kenttä määrittää, saako opiskelija halutessaan nähdä mallivastauksen.

Output-osassa määritetään kertomerkin esitys Multiplication sign -kentässä, neliöjuuren esitys Surd for square root -kentässä (false-arvolla neliöjuuri esitetään murtopotenssina) ja imaginaariyksikön esitys Meaning and display of sqrt(-1) -kentässä.

**Options**

Question Level Simplify: True

Assume Positive: False

Question Penalty: 0.1

Mark Modification: Penalty

Feedback used: Full

Feedback Correct: `<span class='correct'>Correct answer, well done.</span>`

Feedback Partially Correct: `<span class='partially'>Your answer is partially cc</span>`

Feedback Incorrect: `<span class='incorrect'>Incorrect answer.</span>`

Worked solution on demand: True

**Output**

Multiplication Sign: Dot

Surd for Square Root: True

Meaning and display of sqrt(-1): i

**Kuva 3.8.** Tehtävän luomisen käyttöliittymän options-osa.

Question tests -osassa (kuva 3.9) tehtävää voidaan testata automaattisesti. Opettaja määrittää haluamansa määrän vastauksia ans-kenttiin ja merkitsee PotResTree-kenttiin, pitäisikö tarkastuspuiden arvostella ne oikeiksi vai vääriksi. Testien tulokset näytetään tehtävän tarkastelun yhteydessä.

	ans1	PotResTree_1	Remove
0	<input type="text"/>	NONE	<input type="checkbox"/>

Add

**Kuva 3.9.** Tehtävän luomisen käyttöliittymän question tests -osa.

Metadata-osan (kuva 3.10) tärkein osa on publish-kenttä, jonka kautta opettaja voi määrittää, onko tehtävä näkyvillä kaikille STACKia käyttäville opettajille vai vain hänelle itselleen. Jos tehtävä on julkinen, on syytä määrittää myös sen valmiusaste question status -kentällä. Language-kentän käyttöön ei tällä hetkellä ole tarvetta, sillä STACK ei tue suomen kieltä. Targeted to -kentässä määritetään, mille kouluasteelle tehtävä on suunnattu. Difficulty-kentässä määritetään tehtävän vaikeusaste asteikolla erittäin helpposta erittäin vaikeaan. Competencies trained -kentässä määritetään, harjoittaako tehtävä ensisijaisesti ajattelu-, perustelu-, ratkaisu-, esitys-, kieli-, kommunikaatio- vai työkalujen käyttötaitoja. Skills required -kentässä määritetään, vaatiiko tehtävän ratkaiseminen alkeellisia, yksinkertaisia konseptuaalisia, monivaiheisia vai edistyneitä taitoja. Recommended time to take -kentässä määritetään keskimääräinen tehtävän ratkaisuun kuluva aika. Type of question -kenttä kuvaa vastauksen tyyppiä. Sillä ei kuitenkaan ole mitään yhteyttä muihin vastauskenttiin. Rights statement -kenttä määrittää, mistä osoitteesta tehtävän tai järjestelmän käyttöoikeuksista on saatavilla lisätietoa.

Moodle options -osassa (kuva 3.11) tehtävä liitetään yhteen Moodlen tehtäväpankkiin. Tämä vaihe ei ole pakollinen, mutta Moodlen tehtäväpankkien käyttö helpottaa kysymysten ylläpitoa erityisesti silloin, kun niitä on suuri määrä.

► **Metadata** 🗨

Publish: Unpublished ▾

Question Status: Draft ▾

Language: en ▾

Targeted to: Unspecified ▾

Difficulty: Unspecified ▾

Competencies trained: Unspecified ▾

Skills Required: Unspecified ▾

Recommended time to take: 00:00:00

Type of Question: Algebraic Expression ▾

Rights Statement: <http://www.gnu.org/copyleft/gpl.html>

**Kuva 3.10.** Tehtävän luomisen käyttöliittymän metadata-osa.

► **Moodle Options**

You may add this question to one of Moodle's question banks. Select the required category.

Category: Oletus kohteelle Kotitehtävä 2 ▾

Add to Moodle

**Kuva 3.11.** Tehtävän luomisen käyttöliittymän Moodle options -osa.

Ennen kuin tehtävä voidaan esittää opiskelijoille, siitä on vielä luotava instansseja deploy-toiminnolla. Tehtävälle luodaan tietty määrä alkuarvoja, jotka tallennetaan. Tämä nopeuttaa tehtävien esitystä opiskelijoille ja helpottaa niiden testaamista.

### 3.2.3 Tehtävän tallennus

Tietokoneohjelman käyttäjällä ei yleensä ole tarvetta tietää, millaisessa muodossa ohjelma käsittelee tietoa sisällään. STACKin tapauksessa tämä tieto saattaa kuitenkin olla koko järjestelmän mielenkiintoisin tekijä. STACKin suurimmat vahvuudet ovat sillä luotujen matematiikan tehtävien ilmaisuvoima ja monipuoliset tarkastusvaihtoehdot, ei niinkään käyttöliittymä. Tehtävien tallennusmuodon tunteminen mahdollistaa uusien käyttöliittymien luomisen. Tampereen teknillisellä yliopistolla onkin jo luotu uusia opiskelijan käyttöliittymiä sekä MathBridge-projektin yhteydessä että matematiikan perustaitojen testiä varten. Myös uusien opettajan käyttöliittymien luominen on mahdollista. Tehtävien erottaminen sovelluksen muusta sisällöstä tarjoaa myös helpon keinon integroida STACK muihin oppimisympäristöihin tai järjestelmiin.

Tehtävät tallennetaan MySQL-tietokantaan. Tietokantaan tallentuu kolmenlaista dataa:

- 1) dataa tehtävästä,
- 2) väliaikaista dataa tehtävästä ja
- 3) dataa opiskelijan vastauksesta.

Kun opettaja tallentaa luomansa tehtävän, tehtävään liittyvä data tallennetaan taulukossa 3.2 kuvattuihin tauluihin. On huomattava, etteivät tietokantataulujen sarakkeet vastaa suoraan opettajan käyttöliittymän kenttiä. Lisäksi joidenkin tietokannan alkioiden sisältö on tietoturvan parantamiseksi enkoodattu ADOdb-PHP-kirjaston avulla. Esimerkiksi tehtävänantoa ei voida lukea suoraan tietokannasta dekodeamatta dataa.

**Taulukko 3.2.** Dataa tehtävästä tallentavat tietokantataulut.

Taulu	Tehtävä
question_keyword	Yhdistää avainsanat tehtäviin.
question_lines	Ylläpitää listaa tehtävien uusimmista versioista.
response_trees	Tallentaa tarkastuspuut ja yhdistää ne tehtäviin.
stackquestion	Tallentaa suurimman osan tehtäviin liittyvästä datasta.

Tehtävään liittyvää dataa ei kuitenkaan haeta edellä mainituista tauluista joka kerta opiskelijan tai opettajan tarkastellessa tehtävää. Kun tehtävä avataan ensimmäisen kerran, osa sen datasta kopioidaan taulukossa 3.3 kuvattuihin välimuistitauluihin, joista tehtävän tiedot haetaan seuraavilla avauskerroilla. Välimuistitaulujen rakenne ei ole kuitenkaan identtinen varsinaisten tehtävätaulujen kanssa. Esimerkiksi CAStext muunnetaan välimuistitauluja varten XHTML:ksi, jotta sitä ei tarvitse käsitellä uudelleen välimuistista hakemisen jälkeen. Kun tehtävän yrityskerran tilassa tapahtuu muutoksia, esimerkiksi opiskelijan tehtävästä ansaitseman pistemäärän kasvaessa tai vähentyessä, välimuistitauluun luodaan uusi rivi.

**Taulukko 3.3.** Väliaikaista dataa tehtävästä tallentavat tietokantataulut.

Taulu	Tehtävä
display_cache	Tallentaa väliaikaisesti tehtävään liittyvän datan.
display_cache_sequence	Tallentaa display_cache-rivien järjestyksen.

Opiskelijan vastauksiin liittyvät taulukossa 3.4 kuvatut tietokantataulut. Tehtävänannoista poiketen vastaukset tallennetaan tietokantaan puhtaana tekstinä. Myöskään vastaustauluja ei päivitetä, vaan vastaustauluun lisätään uusi rivi opiskelijan muuttaessa vastaustaan.

Lisäksi STACK tallentaa kysymyksissä käytetyt avainsanat keywords-tauluun ja Opaque-protokollan käyttämät istuntotiedot opaque\_sessions-tauluun.

**Taulukko 3.4.** Dataa vastauksesta tallentavat tietokantataulut.

Taulu	Tehtävä
attempt_meta_answer	Tallentaa vastauksen.
attempt_meta_PRT	Tallentaa metatietoa vastauksesta.
question_attempts	Tallentaa käyttäjän id:n ja aikaleiman ja yhdistää ne vastaukseen.

STACK tarjoaa myös mahdollisuuden tehtävän tallentamiseen ja lataamiseen XML-muodossa. Tämä on helppo tapa siirtää tehtäviä yhdestä STACK-järjestelmästä toiseen. XML-tiedostoja ei käytetä muuhun tarkoitukseen kuin tehtävien siirtämiseen STACKista toiseen. Niillä ei siis esimerkiksi siirretä tehtäviä STACKin ja Moodlen välillä. STACK1 ja STACK2 käyttävät hieman erilaisia XML-skeemoja, mutta STACK1:n tehtäviä voidaan siirtää STACK2:een käyttämällä STACK2:een sisältyvää muuntotyökalua. STACK3:n käyttämä XML-skeema poikennee taas edellisistä, mutta kehittäjät ovat luvanneet siihenkin muuntotyökalun (Sangwin 2012).

STACK-tehtävän käyttämä XML-skeema on saatavilla STACK-asennuspaketissa. Pitäen tukiä siitä esitellään tässä diplomityössä vain pääkohdat. STACK-tehtävää kuvaava XML-dokumentti koostuu juurielementistä ja kuudesta pääelementistä, jotka esitetään taulukossa 3.5.

**Taulukko 3.5.** STACK-tehtävää kuvaavan XML-dokumentin pääelementit.

Elementti	Sisältö
questionCasValues	Suurin osa tehtävään liittyvästä datasta: Tehtävän käyttämät muuttujat, tehtävänanto, malliratkaisu ja tehtävän tunniste (question note).
questionparts	Tehtävään liittyvät vastauskentät ja oikeat vastaukset.
PotentialResponseTrees	Tehtävään liittyvät tarkastuspuut.
MetaData	Dublin Core® -metadata, IEEE Learning Object Meta-Data (LOM) ja muu tehtävään liittyvä metadata.
ItemOptions	Tehtävänluonnin Options-osassa määritetty data.
ItemTests	Tehtävään liittyvät automaattiset testit.

Helpoin tapa viedä STACK-tehtäviä muihin ympäristöihin on valita tästä XML-skeemasta sopiva osajoukko, jonka toiminnallisuuden uusi ympäristö lupaa toteuttaa. Tällöin sopivat STACK-tehtävät voidaan viedä uuteen ympäristöön ja kaikki uudessa ympäristössä luodut tehtävät voidaan tarvittaessa siirtää STACKiin. Jos uusi ympäristö

tarjoaa STACKia enemmän toiminnallisuutta, XML-skeema voidaan määrittää myös STACKin tehtäväskeeman ylijoukoksi.

### 3.2.4 Matematiikkaohjelmisto

Matematiikkaohjelmisto eli matematiikkamoottori (englanniksi computer algebra system, CAS) pystyy käsittelemään symbolisia lausekkeita matemaattisin työkaluin. Ne ovat usein suuria, monipuolisia ohjelmia, joiden käyttö ei rajoitu tehtävien tarkastamiseen. Vaikka automaattisesti tarkastettavien tehtävien luominen on mahdollista myös ilman matematiikkaohjelmistoa, sellaisen käyttö helpottaa järjestelmän luojan työtä huomattavasti. Matematiikkaohjelmisto tarjoaa käyttöön suuren määrän operaatioita matematiikan eri osa-alueilta, jolloin niitä ei tarvitse toteuttaa erikseen.

STACK käyttää ilmaista, avoimen lähdekoodin matematiikkaohjelmisto Maximaa, joka on toteutettu Lisp-ohjelmointikielellä. Maximaa voidaan ajaa useilla alustoilla, ja siihen on saatavilla myös graafisia käyttöliittymiä. Maximaa kehitetään edelleen aktiivisesti, mikä voi aiheuttaa pieniä ongelmia, sillä kaikki Maximan versiot eivät ole yhteensopivia kaikkien STACKin versioiden kanssa. STACK ei tue muita matematiikkaohjelmistoja kuin Maximaa. Vaikka ohjelmiston vaihto olisi periaatteessa mahdollista, Maxima on integroitu STACKiin niin vahvasti, että toimenpide olisi hyvin työläs ja edellyttäisi suuria muutoksia myös jo luotuihin tehtäviin. STACK sisältää myös Maximan omalla ohjelmointikielellä luotuja pieniä ohjelmia muun muassa kuvien luomiseen.

Tehtäviä luodessaan opettaja voi syöttää osaan kentistä CASTextiä, joka sisältää Maximan funktioita. Maxima-komentoja syntyy myös tehtävänluonnin muissa osissa, kuten useimmissa vastaustesteissä. Kun STACK havaitsee tehtävässä Maxima-komennon, se käynnistää palvelimella Maximan, ajaa komennot ja ottaa vastaan palautteen.

### 3.2.5 Moodle-integraatio

Moodle (Moodle 2012a) on ilmainen avoimen lähdekoodin oppimisympäristö. Kehittäjät julkaisevat ohjelmasta uuden version kahdesti vuodessa. Keväällä 2012 uusin julkaistu versio on Moodle 2.2 (Moodle 2012c). Tämä luku käsittelee kuitenkin vanhemmaa versiota 1.9, joka on ainoa STACK2:n kanssa yhteensopiva.

STACK yhdistetään Moodleen STACK2:n asennuspakettiin kuuluvan Opaque-moduulin avulla. Moduuli on yleensä kolmannen osapuolen tuottama lisäosa, joka toteuttaa järjestelmään uutta toiminnallisuutta. Opaque-kysymystyyppin lisäksi moduuli sisältää STACK-lohkon. Lohko on Moodlen nimitys sivulle lisättävälle pienelle laatiolle. STACK-lohko sisältää linkkejä STACKin opettajan käyttöliittymään ja on näkyvissä vain kurssin opettajille.



Kysymystyyppi on Moodlen sisäinen malli, johon kuuluu sekä tietosisältöä että toiminnallisuutta. Kysymystyyppi määrittelee, kuinka tehtävänanto ja vastauksen palautus ja mahdollinen koneellinen tarkastus suoritetaan. Moodle 1.9 sisältää kymmenen sisäänrakennettua kysymystyyppiä. Uusien kysymystyyppien luominen edellyttää omaa ohjelmointia. Kolmannen osapuolen toteuttamia kysymystyyppiä on olemassa melko vähän (Moodle 2012b).

Opaque on Tim Huntin kehittämä SOAP-pohjainen protokolla, joka mahdollistaa Moodlen tentti-työkalun keskustelun erillisen kysymysmoottorin kanssa. Opaque kykenee keskustelemaan STACKin lisäksi myös Open Universityn OpenMark-arviointijärjestelmän kanssa. (Hunt 2011) Opaquen käyttö mahdollistaa STACKin sijoittamisen eri palvelimelle Moodlen kanssa. Opaque-moduulille annetaan STACKin osoite ja STACKin asennuksen yhteydessä luotu tunniste, jonka jälkeen Moodle pystyy kommunikoimaan STACKin kanssa SOAP-protokollan avulla.

### 3.3 Arviointia

Luku 2.3 esitteli Bokhoven ja Drijversin suunnitteleman arviointityökalun matematiikan tehtäviä automaattisesti arvioiville järjestelmille. Ennen varsinaista arviointia järjestelmille esitettiin neljä ehtoa. STACK täyttää näistä kolme.

STACK tukee matemaattista esitystä ja algebrallisia operaatioita melko hyvin. Matemaattisten merkintöjen esitys toteutetaan oletusarvoisesti JavaScript-kirjaston avulla. Menetelmä ei ole täysin aukoton, mutta toimii suurimmassa osassa opiskelijoiden käyttämistä selaimista. STACKin vastaustyytit eivät rajoitu numeroarvoihin tai monivalintoihin, vaan STACK pystyy ottamaan vastaan ja tarkastamaan matemaattisen lausekkeen.

STACK tarjoaa opettajalle mahdollisuuden omien tehtävien laatimiseen. STACKin ilmaisuvoima riittää useimpien yliopistomatematiikan peruskurssien sisältöjen esittämiseen. Aivan kaikkien matematiikan tehtävien tarkastaminen ei kuitenkaan onnistu. Tämä johtuu osittain STACKista itsestään, osittain sen käyttämästä Maxima-matematiikkaohjelmistosta. STACKilla voidaan luoda vain sellaisia tehtäviä, jotka on mahdollista tarkastaa Maxima-funktioilla tai STACKin itse tarjoamilla tarkastustoiminnoilla. Hankalia tarkastettavia ovat esimerkiksi todistustehtävät, joissa sekä alkutilanne että haluttu lopputulos annetaan opiskelijalle jo tehtävänannossa ja tarkastuksessa eniten merkitsevät opiskelijan löytämät välivaiheet. Niiden koneelliseen tarkastamiseen ei ole yleistä menetelmää, vaikka joitain yksittäisiä todistustehtäviä onkin luotu STACKille (Ruokokoski 2009). Myöskään esimerkiksi tehtävänannon muuttaminen tehtävän suorituksen aikana ei ole STACKissa mahdollista. Useampiosaisessa tehtävässä tehtävän eri osien vastaukset eivät voi vaikuttaa muiden osien tehtävänantoihin. Esimerkiksi matriisitehtävässä ei ole mahdollista kysyä ensin opiskelijalta tulomatriisin dimensioita ja

luoda sitten vastauksen mukaista matriisia, johon opiskelija täyttäisi oikeat arvot. On olemassa järjestelmiä, joissa tehtävänannon muuttaminen on mahdollista, kuten Maple T.A., jossa opiskelijalle voidaan virheellisen vastauksen jälkeen näyttää samankaltainen tehtävä helpotetuilla alkuarvoilla tai sama tehtävä vaiheistettuna.

STACK tallentaa opiskelijoiden vastaukset ja mahdollistaa opettajalle niiden tarkastelun. Käytännössä arvostelujen tarkastelun käyttöliittymä on STACK2:ssa jätetty Moodlen vastuulle, mikä on hyvä valinta. Opettaja voi helposti yhdistää STACK-tehtävien pisteet ja arvosanat muihin kurssin Moodlella arvosteltuihin osasuorituksiin. STACK esittää tehtävän viimeisen vastausyrityksen helposti ymmärrettävässä muodossa, mutta aiempien vastausyritysten tulkinta ei valitettavasti ole intuitiivista.

Vaatus teknisen tuen saatavuudesta täyttyy vain osittain. STACK2:n kehitys on jo lopetettu, mikä herättää kysymyksen, onko perusteltua käyttää järjestelmää, johon ei enää ole saatavilla edes virheitä korjaavia päivityksiä ja joka todennäköisesti korvautuu lähiaikoina uudemmalla versiolla. Kehityksen loppuminen herättää kysymyksiä myös tulevaisuuden yhteensopivuudesta. Toisaalta seuraavan version valmistumisen ajankohdasta tai paremmuudesta ei ole takeita, eikä sen odottaminen ole välttämättä järkevää. StackWiki-sivustolle on dokumentoitu STACK2:n asennus sekä opiskelijan ja opettajan käyttöliittymät (StackWiki 2011). Dokumentaatio ei kuitenkaan ole täydellinen, eivätkä kaikki ominaisuudet toimi aivan kuten dokumentaatioissa.

Bokhove ja Drijvers arvioivat STACK2:n seitsemästä tarkastelemastaan järjestelmästä neljänneksi parhaaksi. He kiittivät STACK2:ta tarkastuspuiusta, moniosaisista kysymyksistä ja integraatiosta oppimisympäristöön, mutta moittivat asennuksen hankaluutta, vakautta, suorituskykyä, ulkonäköä ja jatkuvuuden puutetta. (Bokhove & Drijvers 2010) Tampereen teknillisen yliopiston kokemukset ovat samansuuntaisia.

STACK-tehtäviä on käytetty niin itsenäisessä sovelluksessa kuin Moodle- ja Active-Math-oppimisympäristöissä. Tästä voidaan päätellä, että STACK-tehtävät sisältävät arvokasta tietoa, ne on esitetty toimivassa formaatissa ja niiden integrointi muihin järjestelmiin ei vaadi kohtuutonta vaivaa. STACKilla luotuja tehtäviä voitaisiin siis hyödyntää tulevaisakin järjestelmissä. Matemaattisten tehtävien esittämiseksi ei ole toistaiseksi olemassa standardeja, vaikka yhtenäinen tapa mahdollistaisi tehtävien siirtämisen ympäristöstä toiseen.

STACKin asennusta vaikeuttavat erityisesti sen riippuvuudet useisiin muihin ohjelmiin. Joistakin ohjelmista, kuten matematiikkaohjelmisto Maximasta, kelpaavat vain tietyt versiot. Erityisen merkittävä on STACK2:n käyttämä Moodlen versio. Ainoa yhteensopiva versio on Moodle 1.9, joka julkaistiin vuonna 2008 ja jonka varsinainen kehitys on jo lopetettu. Kehittäjät ovat ilmoittaneet tekevänsä versioon 1.9 vain kriittisiä tietoturvapäivityksiä ja niitäkin vain kesäkuuhun 2012 asti (Moodle 2012c). Moodlen käyttäjil-

le suositellaan puolivuositain päivittyvään versiohaara 2:een siirtymistä. On yleisesti suositeltavaa käyttää ohjelmistojen uudempia versioita parempien ominaisuuksien ja tietoturvan takia. STACK2:n käyttö pakottaa pysymään vanhassa Moodle-versiossa.

Myös STACK2:ta ajavan palvelimen asetuksia on säädettävä STACKille sopiviksi: PHP-varoituksia ja -huomautuksia ei ole syytä pitää päällä edes kehityspalvelimella, sillä STACK2:n admin-käyttöliittymä tuottaa niitä niin runsaasti, ettei järjestelmää pysty käyttämään normaalisti. Maximan ajon optimointi on välttämätöntä, jotta STACK saadaan toimimaan kohtuullisella nopeudella.

Käytössä havaittiin Opaquen käyttämän SOAP-protokollan tuottavan ajonaikaisia virheitä, jotka vaikeuttavat erityisesti STACKin opettajan käyttöliittymän käyttöä Moodlen kautta. Suomenkielisissä tehtävissä havaittiin ongelmia myös skandinaavisten merkkien koodauksessa tietokantaan ja tehtävien XML-esityksissä.

Luvussa 4.2 kuvatuissa testeissä havaittiin, että opiskelijoilla on vaikeuksia ymmärtää STACK-tehtävien käyttöliittymää. Vaikka moniosaiset tehtävät voivat tarjota opettajalle enemmän tietoa opiskelijan taidoista ja laskutekniikasta, niiden validoiminen ja tarkastaminen on opiskelijoille haastavaa. STACKista puuttuu mekanismi, joka kertoisi opiskelijalle yksiselitteisesti, onko hän jo tarkastuttanut kaikki tehtävän osat.

Taulukko 3.6 vertaa STACK2:ta luvussa 2.6 esitettyihin vaatimuksiin. STACK2:n ongelmat ovat tässäkin vertailussa tekninen toimivuus ja opiskelijan käyttöliittymä. Hintalaatu-suhteen arviointi on vaikeaa.

**Taulukko 3.6.** STACK2:n arviointi luvun 2.6 vaatimuksilla.

Vaatus	Suoriutuminen
Yliopistotasoiset tehtävät	Hyvä
Riittävä suorituskyyky ja tekninen toimivuus	Heikko
Opiskelijoiden hyväksyntä	Heikko
Mahdollisuus uusien tehtävien laatimiseen	Hyvä
Vastausten ja arvostelujen tallennus	Riittävä
Hyvä hinta-laatu-suhde	?
Integroituvuus muihin käytössä oleviin järjestelmiin	Hyvä

STACKin käytöstä muualla saadut kokemukset ovat kuitenkin olleet lupaavia. Aalto-yliopisto on käyttänyt STACKia jo vuosia. Sekä OtaSTACKin että STACK2:n on osoitettu toimivan suurillakin kursseilla. Aalto-yliopistossa on myös havaittu opiskelijoiden suhtautuvan STACK-tehtäviin myönteisesti ja kurssiarvosanojen korreloivan suoritettujen STACK-tehtävien kanssa (Rasila et al. 2010). Lisäksi STACKin kokeileminen on helppoa. Ilmaiseen, avoimen lähdekoodin järjestelmään on mahdollista tutustua nopeasti ja sitoutumatta, toisin kuin kaupallisiin ohjelmistoihin, joiden käytöstä on tehtävä päätös jo ennen lisenssineuvottelujen aloittamista.

## 4 STACK KÄYTÖSSÄ

### 4.1 Tehdyt muutokset

Opettajan ohjeissa esitetyt asennusohjeet perustuvat asennuspakettiin, joka on hieman muunneltu versio STACK2:n kehitysversiosta, jonka viimeisimmät kehittäjän tekemät muutokset julkaistiin 22.1.2012. Asennuspaketin merkittävin ero viralliseen jakeluun on jsMath-JavaScript-kirjaston vaihto MathJaxiin. Lisäksi Maximaa on optimoitu toimimaan paremmin TTY:n palvelimella.

Asennusohjeissa oletetaan myös, että käytössä on Tampereen teknillisellä yliopistolla muokattu versio Moodle 1.9:stä. Tämän version merkittävin ero viralliseen jakeluun on se, että TTY:n Moodle tukee Shibboleth-kertakirjautumisjärjestelmää. Kertakirjautumisjärjestelmän ansiosta opettajien ja opiskelijoiden ei tarvitse luoda Moodleen erillisiä tunnuksia, vaan he voivat käyttää sitä TTY:n intranet-tunnuksillaan. Jotta STACK pystyisi esittämään matemaattiset merkinnät oikein, myös Moodlen käytössä oleviin teemoihin on sisällytettävä MathJaxin lataus ja alustus.

### 4.2 Testaus

STACKia testattiin Tampereen teknillisellä yliopistolla kevätlukukaudella 2012 opintojakson Insinöörimatematiikka B 4u yhteydessä. Kyseessä on ensimmäisen vuosikurssin diplomi-insinööriopiskelijoille suunnattu opintojakso, joka käsittelee usean muuttujan reaali- ja vektoriarvoisia funktioita, ääriarvoja sekä taso- ja avaruusintegraaleja. Insinöörimatematiikka 4u -opintojaksolla on neljä rinnakkaistoteutusta, joista B on suunnattu, muttei rajoitettu, tuotantotalouden, signaalinkäsittelyn ja tietoliikennetekniikan ja sähkötekniikan koulutusohjelmien opiskelijoille. Opintojaksolle ilmoittautui 259 opiskelijaa.

Opintojakso koostui 13 luennosta, 7 laskuharjoituksesta, 3 tietokoneharjoituksesta sekä tentistä. Läpäistäkseen opintojakson opiskelijan oli saatava tentistä hyväksytty arvosana, suoritettava vähintään 40 % laskuharjoitustehtävistä ja osallistuttava vähintään kahteen tietokoneharjoitukseen.

STACK-tehtäviä kokeiltiin yhden tietokoneharjoitusryhmän yhdessä tietokoneharjoituksessa sekä kaikkien opiskelijoiden saatavilla olevina Moodlen kautta jaeltuina bonuspistetehtävinä. Molemmissa testeissä testattiin STACK2:ta, jota ajettiin kehityspalvelimella Moodle-oppimisympäristön kautta. Myös tehtävien luomista OtaSTACKilla

kokeiltiin, mutta nämä testit olivat pienimuotoisempia eivätkä liittyneet mihinkään opintojaksoon. OtaSTACKia kokeilivat TTY:n opettajat.

STACK2:n testauksen pääasialliset tavoitteet olivat:

- selvittää, onko STACK TTY:lle sopiva järjestelmä
- selvittää, minkälaisia tehtäviä STACKilla on mahdollista luoda, ja
- vertailla STACKin eri versioita keskenään.

Testeissä huomattiin, että STACKin opiskelijan käyttöliittymässä on merkittäviä ongelmia sekä toiminnallisuudessa että käytettävyydessä. Tehtäviä luodessa paljastui uusia rajoituksia, mutta STACKin todettiin niistä huolimatta soveltuvan yliopistotasoisien matematiikan tehtävien luomiseen. STACK2:n Moodle-integraatiota pidetään edelleen merkittävänä etuna, vaikka OtaSTACKin opiskelijan käyttöliittymä onkin hieman intuitiivisempi.

Testeissä ei päästy selvittämään, kuinka opettajat käyttävät STACKia tehtävien luomiseen, tai millainen vaikutus opettajan ohjeilla on STACKin käyttöön. Jos STACK otetaan laajempaan käyttöön, näitäkin asioita on syytä tutkia.

#### 4.2.1 Testi 1

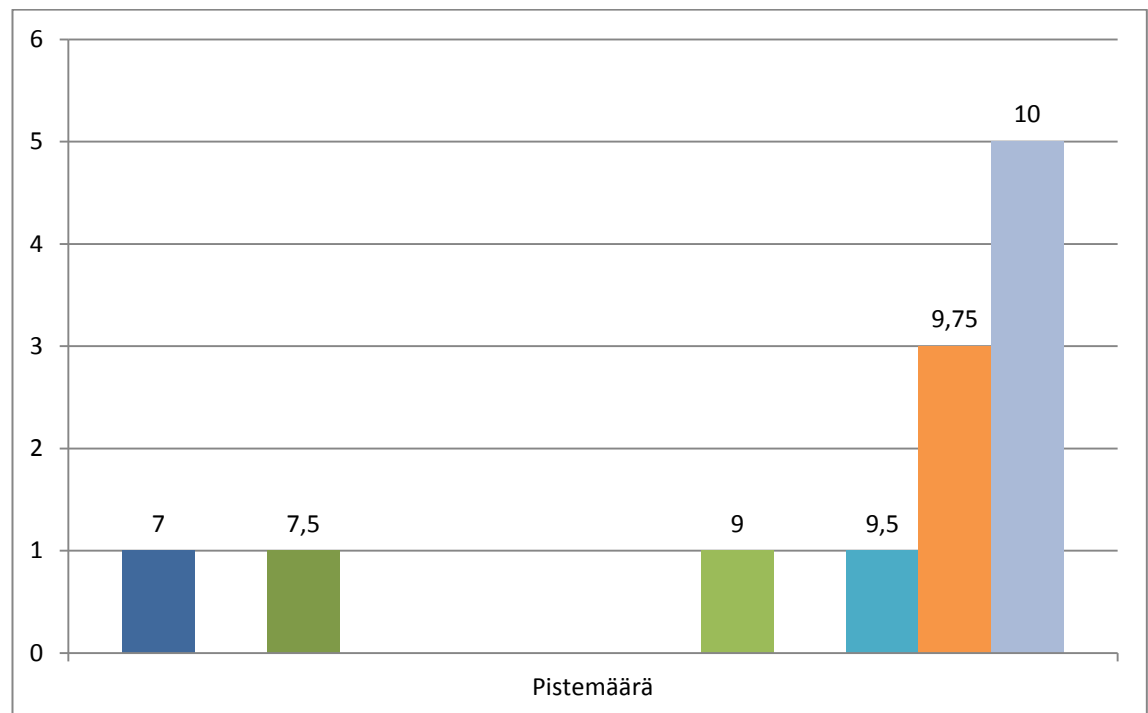
Ensimmäinen testi suoritettiin eräessä Insinöörimatematiikka B 4u -opintojakson tietokoneharjoitustilaisuudessa. Tavallisesti opiskelijat suorittavat tietokoneharjoituksessa tehtäviä MatLab-matematiikkaohjelmistolla. Useimmat opiskelijat ratkaisevat tehtävät kolmen opiskelijan ryhmissä, mutta myös tehtävien tekeminen pareittain tai yksin voi olla mahdollista tilaisuuden osallistujamäärästä riippuen. Opiskelijat saavat käyttää harjoituksessa apuvälineitä, kuten laskimia ja tietokoneohjelmia, ja materiaalia, kuten kurssin luentomonistetta tai oppikirjaa tai Internet-resursseja.

Testiin osallistuivat harjoitustilaisuudessa läsnä olleet 30 opiskelijaa, jotka työskentelivät 13 1-4 hengen ryhmässä. Tuloksissa näitä ryhmiä kutsutaan vastaajiksi erotuksena yksittäisistä opiskelijoista. Testissä vastaajilla oli noin viisitoista minuuttia aikaa tehdä kaksi STACK-tehtävää, jonka jälkeen heille jäi noin puoli tuntia aikaa MatLab-tehtäville. Kaikki opiskelijat saivat harjoituksesta läsnäolomerkinnän ratkaistujen tehtävien määrästä riippumatta. STACK-tehtävistä ensimmäisessä vastaajien oli laskettava annetun käyrän derivaattavektori, toisessa kaksi annetun kolmen muuttujan funktion osittaisderivaattaa. Molemmissa tehtävissä käytettiin satunnaistettuja alkuarvoja. Vastaukset arvosteltiin vastauspuun yhdellä solmulla joko oikeiksi tai vääriksi ilman räätälöityä palautetta. Tehtävien tekemisen jälkeen vastaajat vastasivat lyhyeen kyselyyn, jossa heitä pyydettiin kertomaan mahdollisesti kohtaamistaan ongelmista.

Testin tavoitteet olivat:

- 1) varmistaa, että STACKilla voidaan luoda yliopistotasoisia matematiikan tehtäviä
- 2) selvittää, esiintyykö STACKin käytön aikana teknisiä ongelmia
- 3) antaa viitteitä siitä, kuinka STACK kestää samanaikaista kuormitusta, ja
- 4) selvittää, kuinka opettaja voi hyödyntää STACKin tuottamaa dataa.

Vastaajat käyttivät kahden STACK-tehtävän tekemiseen 10–15 minuuttia. Kumpikin tehtävä koostui kahdesta osasta, joiden maksimipistemäärä oli yksi piste. Jokaisesta väärästä vastauksesta vähennettiin 0,1 pistettä. Koko tehtäväsarjan maksimipistemäärä oli neljä pistettä, jonka Moodle skaalasi kymmeneksi pisteeksi. Vastaajat saivat suorittaa tehtäväsarjan niin monta kertaa kuin halusivat, jolloin lopulliseksi tulokseksi jäi paras saavutettu pistemäärä. Tehtäväsarjan uusi vain yksi vastaaja, joka lopetti epähuomiossa ensimmäisen yrityksensä ennenaikaisesti. Vastaajien pisteet esitetään kuvassa 4.1. Keskiarvo on 9,21 pistettä.



**Kuva 4.1.** Vastaajien pistemäärät tietokoneharjoituksessa suoritettussa STACK-tehtävässä.

Vastaajien tekemistä virheistä suurin osa liittyi matematiikan osaamiseen: Vastaaja esimerkiksi derivoi funktion väärin, derivoi väärän funktion tai ei osannut laskea funktion arvoa tietyssä pisteessä. Toinen virhelähde oli STACKin käyttämä syntaksi. Vastaajat tekivät syntaksivirheitä sekä tehtävänannossa ohjeistetuissa että ohjeistamatta jätetyissä merkinnöissä. Suurin osa syntaksivirheistä paljastui vastaajille vastauksen validoinnin yhteydessä, mutta yksi sisällöltään oikea mutta syntaksiltaan väärämuotoinen vastaus päästettiin läpi ja arvosteltiin virheelliseksi. Vastaajat raportoivat kohdanneensa myös teknisiä ongelmia, mutta ne eivät johtaneet pistevähennyksiin.

Suurin osa vastaajista kertoi, ettei pidä STACK-tehtäviä yhtä hyödyllisinä kuin tavallisia viikkoharjoitustehtäviä, jotka ratkaistaan paperilla ja esitetään harjoitustilaisuuksissa. Useat esitetyistä perusteluista liittyivät tehtävien matemaattiseen helppouteen. Vastaajia epäilytti myös, olisiko STACK-tehtäviin saatavilla tukea.

Tehtävät suunnittelivat tämän diplomityön kirjoittaja ja kurssista vastaava matematiikanopettaja. Tehtävät toteutti kirjoittaja. Vaikka tehtävät suunniteltiin tarkoituksella nopeiksi tehdä, testi osoitti, että STACKilla on mahdollista luoda matematiikan tehtäviä yliopistotasolla käsitellyistä aihealueista ja esittää ne yleisesti käytössä olevalla matemaattisella notaatiolla.

Testin aikana esiintyi pieniä teknisiä ongelmia. Kolme vastaajaa raportoi ongelmista tehtävien välisessä navigoinnissa. Kaikki vastaajat onnistuivat kuitenkin palauttamaan tehtävän hyväksytysti. Yksi vastaaja raportoi järjestelmän toimineen hitaasti. Varsinaisia suorituskykyongelmia ei kuitenkaan esiintynyt. Testin aikana ilmeni myös merkistökoodausongelmia ä- ja ö-kirjainten kanssa, mutta yksikään opiskelija ei raportoinut niitä.

Opiskelijoiden antamia vastauksia tarkastellessa todettiin, että järjestelmä on tallentanut kaikki opiskelijoiden syötteet. Vastauksista on kuitenkin hankala tulkita, mitkä syötteet on validoitu ja mitkä tarkastettu, ja millä perustein opiskelijoiden saamat pisteet jaettiin.

#### **4.2.2 Testi 2**

Toinen testi oli avoin ja vapaaehtoinen kaikille Insinöörimatematiikka B 4u -opintojakson osallistujille. Opintojakson Moodle-sivulle lisättiin linkki kehityspalvelimella sijaitsevaan toiseen Moodle-sovellukseen, jossa opiskelijat pystyivät suorittamaan kolmen STACK-tehtävän tehtäväsarjan. Koko tehtäväsarjan hyväksytysti suorittaneet opiskelijat saivat kolme ylimääräistä laskuharjoituspistettä. Tavallisesti opiskelijat ansaitsivat laskuharjoituspisteitä esittämällä ratkaisemiaan tehtäviä laskuharjoitustilaisuuksissa. Opintojakson hyväksytyä suoritusta varten opiskelijoiden oli kerättävä tietty määrä laskuharjoituspisteitä. Jos opiskelija keräsi vähimmäismäärää enemmän pisteitä, ne lisättiin tentin pistemäärään, jolloin ne saattoivat vaikuttaa kurssin kokonaisarvosanaan korottavasti.

Tehtävien yrityskertojen määrää ei ollut rajattu. Suoritus katsottiin hyväksytyksi, kun opiskelija vastasi kaikkiin tehtäviin oikein. Vääristä vastauksista ei vähennetty pisteitä. Toisin kuin ensimmäisessä testissä, tällä kertaa opiskelijoiden suoritukset olivat henkilö- eivätkä ryhmäkohtaisia. Tehtävien suoritusta ei kuitenkaan valvottu, joten on mahdollista, että opiskelijat saivat apua.



STACK-tehtävistä ensimmäisessä opiskelijoiden oli laskettava annetun kolmen muuttujan funktion gradientti funktiona ja annetussa pisteessä. Toisessa STACK-tehtävässä opiskelijoiden oli laskettava annetun kahden muuttujan vektoriarvoisen funktion derivaattamatriisi. Kolmannessa tehtävässä opiskelijoiden oli muunnettava annettu funktio sylinterikoordinaatteihin, löydettävä sen integrointirajat ja laskettava määrätty integraali. Kaikissa tehtävissä käytettiin satunnaistettuja alkuarvoja. Kaksi ensimmäistä tehtävää arvosteltiin vastauspuun yhdellä solmulla joko oikeiksi tai vääriksi ilman räätälöityä palautetta. Integrointitehtävä koostui kuudesta eri tarkastuksesta: kolmen eri integraalin rajat, muuttujien valinta, integroitava lauseke ja integraalin lopputulos. Näiden tarkastusten tulokset kerrottiin opiskelijalle erillisinä kohtina. Tehtävien tekemisen jälkeen vastaajat vastasivat lyhyeen kyselyyn, jossa heitä pyydettiin kertomaan mahdollisesti kohtaamistaan ongelmista.

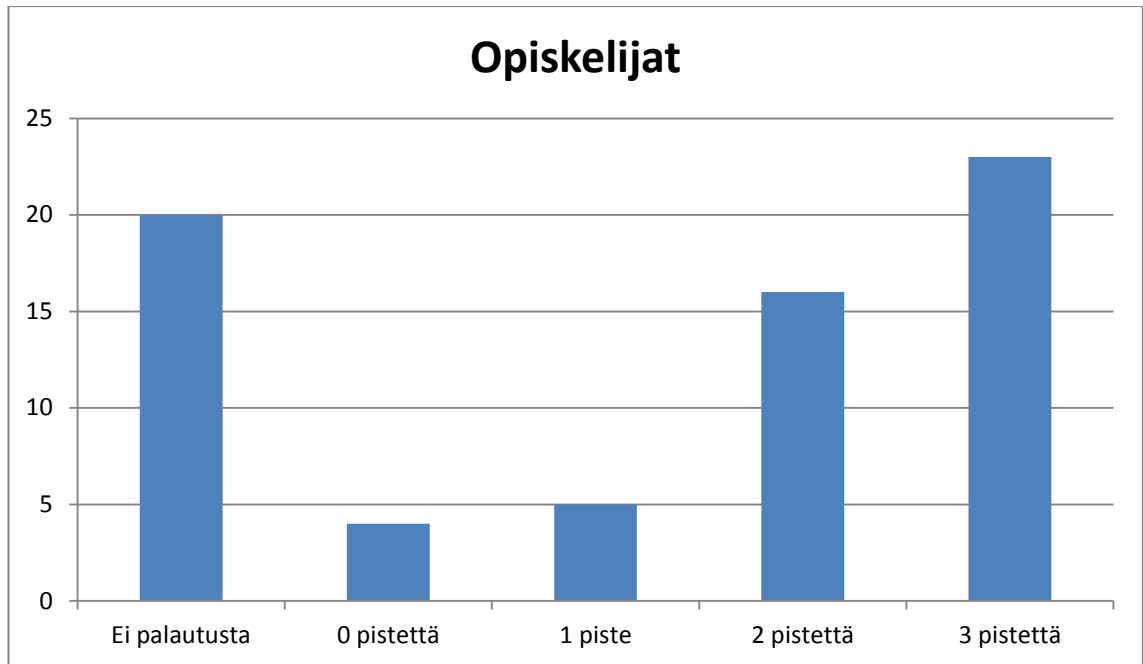
Testin tavoitteet olivat:

- 1) selvittää, esiintyykö STACKin käytön aikana teknisiä ongelmia
- 2) antaa viitteitä siitä, kuinka opiskelijat suhtautuvat STACK-tehtäviin, ja
- 3) selvittää, kuinka opettaja voi hyödyntää STACKin tuottamaa dataa.

Tehtävät avattiin opiskelijoille 3.5.2012 ja ne olivat auki 10.5.2012 asti. Tänä aikana 68 opiskelijaa kirjautui sisään katselemaan STACK-tehtäviä. Heistä 48 palautti tehtäväsarjan. Tehtävien maksimipistemäärät olivat 2, 1 ja 6 pistettä. Koko tehtäväsarjan maksimipistemäärä oli yhdeksän pistettä, jonka Moodle skaalasi kymmeneksi pisteeksi. Keskiarvojen tarkasteleminen ei kuitenkaan ole hyödyllistä, sillä tehtävän todellinen painoarvo ei tässä tapauksessa ole riippuvainen sen maksimipistemäärästä. Vastaajat saivat suorittaa tehtäväsarjan niin monta kertaa kuin halusivat, jolloin lopulliseksi tulokseksi jäi paras saavutettu pistemäärä. Viisi opiskelijaa palautti tehtäväsarjan kahdesti. Neljä opiskelijaa aloitti hyväksytyyn palautuksen jälkeen uuden yrityksen, jota ei vienyt loppuun asti.

Pistemäärien ilmoittamisessa opiskelijoille esiintyi ongelmia. Tehtävän suorituksen aikana opiskelijat eivät tieneet, mikä on maksimipistemäärä. On mahdollista, että osa opiskelijoista kuvitteli yhden pisteen riittävän tehtävän hyväksymiseksi, kun todellisuudessa porkkanapiste myönnettiin vasta, kun opiskelija sai tehtävästä maksimipisteet. Jaetut porkkanapisteet esitetään kuvassa 4.2.

Neljännän tehtävän avoimiin kysymyksiin vastattiin 39 kertaa. Lisäksi yksi opiskelija ilmoitti kokemistaan teknisistä ongelmista sähköpostitse. Teknisiä ongelmia esiintyi matemaattisten merkintöjen näyttämässä, ä- ja ö-kirjainten näyttämässä, tehtävien välillä navigoinnissa ja sivujen hitaassa latautumisessa. Ongelmia koki kuitenkin vain pieni osa vastaajista.



**Kuva 4.2.** STACK-porkkanapistetehtävästä jaetut porkkanapisteeet.

Opiskelijoiden mielipiteet STACKin käytettävyydestä jakautuivat. 62 prosenttia vastanneista ei ilmoittanut kokeneensa ongelmia. Muut vastaajat kertoivat pitävänsä ongelmallisina syötettyjen arvojen esittämistä syntaksin tarkastuksessa, tiettyjen matemaattisten merkintöjen syntaksia, Submit-painikkeen käyttöä kahteen eri tarkastukseen, tarkastuksen tuloksen löytämistä ja ohjeistuksen puutetta. Vastaajista 61 prosenttia piti STACK-tehtäviä yhtä hyödyllisinä kuin tavallisia viikkoharjoitustehtäviä ja 26 prosenttia piti STACK-tehtäviä hyödyllisempinä. STACK-tehtävät miellettiin kuitenkin enemmän viikkoharjoituksia täydentäviksi kuin korvaaviksi tehtäviksi. Opiskelijat kiittivät erityisesti välitöntä palautetta.

Opiskelijoiden palauttamia vastauksia tarkastellessa huomattiin, että raporttien lukeminen ja hyödyntäminen on helpointa, kun kaikilla tentin tehtävillä on sama maksimipistemäärä. Moodle ei kuitenkaan pysty skaalaamaan STACKin tuottamia pisteitä automaattisesti, joten pistemäärä on päätettävä jo tehtävää luodessa. Tämä on käytännössä vaikeaa, sillä samaa tehtävää voidaan käyttää useissa eri tenteissä.

### 4.3 Ylläpitosuunnitelma

STACKia voidaan hyödyntää Tampereen teknillisen yliopiston matematiikanopetuksessa kahdella tavalla: Se voidaan ottaa kokonaisuudessaan käyttöön tässä diplomityössä kuvatulla tavalla, tai siitä voidaan ottaa käyttöön vain osa toiminnallisuudesta. Jos STACK otetaan käyttöön kokonaisuudessaan, on ensimmäiseksi valittava käytettävä versio. Vaihtoehdot ovat STACK1, OtaSTACK, STACK2 sekä kehitteillä oleva STACK3. Kuten luvussa 3 perusteltiin, STACK2 vaikuttaa vaihtoehdoista parhaalta. Luvussa 4 esitellyt opettajan ohjeet voidaan ottaa käyttöön sellaisinaan.

STACK2:n kaikki ominaisuudet eivät kuitenkaan ole toivottuja. Opiskelijan käyttöliittymää pidetään sekavana ja hankalana käyttää ja sekä Maxima- että Moodle-integraatiot voitaisiin toteuttaa tehokkaammin. Tampereen teknillisellä yliopistolla on jo aiemmin toteutettu samankaltaisia pienempiä järjestelmiä. On siis mahdollista ottaa STACKista käyttöön vain opettajan käyttöliittymä, jonka luomat tehtävät siirretään XML-muodossa toiseen järjestelmään, jossa voidaan käyttää itse luotua opiskelijan käyttöliittymää ja/tai tehtävämoottoria. Toinen järjestelmä voi olla integroitu Moodleen tai muuhun oppimisympäristöön. Opettajan ohjeet voidaan tällöin ottaa käyttöön pienin muutoksin.

Tässä ylläpitosuunnitelmassa oletetaan, että STACK2 otetaan käyttöön asennuspaketissa olevassa muodossa ja että useat opettajat käyttävät sitä useilla matematiikan opintojaksoilla. STACK-tehtäviä käytetään opetuksessa ylimääräisinä bonuspistetehtävinä, vaihtoehtoisina viikkoharjoitustehtävien kanssa tai vapaaehtoisina lisätehtävinä. Oletetaan kuitenkin, että opintojakso on mahdollista läpäistä myös STACK-tehtäviä tekemättä, jolloin järjestelmän saavutettavuudelle ja toimintakyvylle asetettavat vaatimukset jäävät hieman pienemmiksi.

Koska opettajilla ei toistaiseksi ole kokemusta STACKin käytöstä ja virheiden korjaaminen ei käytännössä ole mahdollista sen jälkeen, kun opiskelijat ovat aloittaneet tehtävien tekemisen, tehtävien luominen ja testaus on järkevää suorittaa erillisessä ympäristössä. Tietoturva- ja suorituskysyistä STACK ja Moodle kannattaa tuotantokäytössä erottaa eri palvelimille. STACK-ympäristö koostuu siis seuraavista osista:

1. Kehitys-STACK
2. Kehitys-Moodle
3. Tuotanto-STACK
4. Tuotanto-Moodle

Koska STACK2:ssa on paljon riippuvuuksia muihin ohjelmiin, päivitysten asennuksen yhteydessä on noudatettava erityistä varovaisuutta. STACKiin voivat vaikuttaa Maximan, Moodlen ja MathJaxin päivitykset. On myös mahdollista, että jokin taho tuottaa vielä STACK2:een tietoturvapäivityksiä, vaikka versiohaaran varsinainen kehitys on jo lopetettu. Järjestelmää päivitettäessä on syytä seurata vakiintunutta käytäntöä asentaa päivitykset ensin kehitysjärjestelmään ja siirtää ne tuotantokäytössä olevaan järjestelmään vasta, kun on varmistuttu, että kaikki toiminnot ja vanhat tehtävät toimivat edelleen halutusti.

Opettajat luovat uudet STACK-tehtävät kehitys-Moodlen kautta. Kun tehtävä on valmis ja testattu, opettaja siirtää sen tuotanto-Moodleen, jonka jälkeen hän voi sisällyttää tehtävän tentteihin valitsemalla sen tehtäväpankista. Opettajilla ei ole suoraa pääsyä kummankaan STACKin ylläpitäjän käyttöliittymään, vaan he käyttävät vain Moodleja.

Kysymyspankkien ryvettymistä ehkäistään nimeämis- ja avainsanakäytännöillä. Koska eri matematiikan opintojaksojen sisällöt ovat kohtalaisen eriytyneitä, tuotanto-Moodlessa Ydinjärjestelmä-kysymyspankin alle luodaan oma kysymyskategoria jokaiselle opintojaksolle, ja opintojaksoon liittyvät tehtävät osoitetaan siihen. On huomattava, ettei tehtäviä osoiteta tiettyyn Moodle-kurssiin, sillä samoja tehtäviä saatetaan käyttää eri rinnakkaistoteutuksissa ja eri lukuvuosina. Kehitysjärjestelmässä opettajia kannustetaan pitämään keskeneräiset tehtävät yksityisinä, dokumentoimaan julkaistut keskeneräiset tehtävät ja testaamaan valmiit tehtävät huolellisesti. Näistä toimenpiteistä huolimatta on todennäköistä, että kehitys-STACKin tehtävät on käytävä läpi säännöllisin väliajoin, esimerkiksi kerran vuodessa. Tarkastuksen yhteydessä hylätyt toimimattomat tehtävät poistetaan ja aktiivisessa kehityksessä olevien tehtävien avainsanat ja nimet tarkastetaan.

## 5 OPETTAJAN OHJEET

STACKin kehittäjät ovat keränneet dokumentaatiota järjestelmästä erilliselle sivustolle (StackWiki 2011a). Suurin osa siitä on kuitenkin pelkkää toimintojen ja ominaisuuksien kuvailuja. Tampereen teknillisellä yliopistolla päätettiin kirjoittaa STACKin käyttöä kokeileville opettajille erillinen ohje, joka esittelee parhaita käytäntöjä STACKin käyttöön. Tämä luku määrittää ohjeiden kohderyhmän, perustelee niiden tarpeen, kertoo niiden toteutuksesta ja esittelee niiden sisällön.

### 5.1 Opettajan rooli

Opettajan työtehtävät vaihtelevat suuresti opetusta järjestävän instituution, käytettävissä olevien resurssien ja opettajan omien mielenkiinnonkohteiden mukaan. Alakoulun matematiikanopettajan oletetaan seuraavan kansallista opetussuunnitelmaa ja kaupallisesti tuotettua oppikirjaa, yliopiston syventävän opintojakson matematiikanopettaja saa huomattavasti vapaammat kädet opetuksensa sisällön suhteen. Joskus opettaja hoitaa kaikki opintojaksoon liittyvät järjestelyt yksin, joskus opintojakson toteuttamiseen osallistuu useampi luennoija ja joukko assistentteja.

Myöskään verkko-opintojakson opettajan rooli ei ole vakiintunut. Tietokoneohjelmoinnista innostunut opettaja on voinut luoda opintojaksonsa kotisivulle monenlaisia oppimisaihioita, vähemmän viehättynyt on tyytynyt jakamaan luentokalvonsa verkossa oppimisympäristön valmista työkalua käyttäen. Luodakseen automaattisesti tarkastettavia matematiikan tehtäviä opettajan on hallittava järjestelmän käyttämän matematiikkaohjelmisto perusteellisesti. Joissakin järjestelmissä opiskelijan vastauksen tarkastusprosessin määrittely edellyttää opettajalta myös ohjelmointitaitoa.

Yksi STACKin kehitystä ohjanneista periaatteista oli, ettei opettajan tarvitse opetella ohjelmointia tehtävien tarkastusta vasten. Järjestelmä on onnistunut siinä osittain. Tarkastuspuun luonti onnistuu lomakkeella, mutta tehtävän rungossa ja mallivastauksessa voidaan joidenkin tehtävien yhteydessä joutua käyttämään keinotekoisesti luotuja ehtoja silmukkarakenteita. Myös tehtävien parametrisoinnissa käytetyt muuttujat muistuttavat ohjelmointikielissä esiintyviä. Lisäksi opettajan käyttöliittymä sisältää paljon erilaisia toimintoja, joihin tutustumiseen menee aikaa. Nämä voivat olla syitä sille, ettei STACK ole vielä yleistynyt joka opettajan perustyökaluksi.

Aalto-yliopisto on ratkaissut jyrkän oppimiskäyrän ongelman opettamalla STACK-tehtävien luomisen assistenteille. Kun opettaja haluaa ottaa opintojaksolleen STACK-

tehtäviä, hän suunnittelee ne apunaan STACKiin perehtynyt assistentti, joka sitten toteuttaa tehtävän. Luvussa 4.2 kuvatut Tampereen teknillisellä yliopistolla järjestetyt STACK-testit luotiin samalla tavalla: Matematiikanopettaja suunnitteli tehtävät, jotka työn kirjoittaja toteutti.

Tällainen järjestely vähentää huomattavasti opettajan työtä, sillä hänen ei tarvitse opetella STACKin käyttöä lainkaan. STACK-assistentit sen sijaan käyttävät järjestelmää paljon ja harjaantuvat taitaviksi käyttäjiksi. Heitä ei sidota yhteen opintojaksoon, vaan he voivat siirtää STACKiin minkä tahansa opintojaksojen tehtäviä. Tällainen järjestely vaatii kuitenkin henkilöresursseja: kymmenen opintopisteen opintojakson tehtävien luominen vaatii yhdeltä henkilöltä kolmen kuukauden työpanoksen (Rasila et al. 2010). Lisäksi tehtäviä suunnittelevat opettajat eivät välttämättä ymmärrä STACKin koko potentiaalia, koska he eivät osallistu tehtävien toteuttamiseen.

Opettajan ohjeita luodessa tunnistettiin kolme STACKin käyttöön liittyvää tehtävää: Järjestelmän ja palvelimen tekninen ylläpito, STACK-tehtävien luominen ja Moodle-kursseilla avustaminen. Nämä tehtävät on järkevää jakaa eri henkilöille:

- Ylläpitäjä on tietotekniikan ammattilainen. Hän huolehtii STACK-järjestelmän ja –palvelimen toiminnasta, mutta ei osallistu tehtävien tekemiseen.
- Opettaja on matematiikan ammattilainen. Hän suunnittelee tehtävät ja käyttää STACKia niiden luomiseen. Hän luo myös opintojakson Moodle-kurssin.
- Assistentti avustaa opintojakson toteutuksessa. Hän ei voi luoda uusia STACK-tehtäviä, mutta hän voi jakaa opiskelijoille muiden luomia tehtäviä, tarkastella opiskelijoiden vastauksia. Hänellä voi olla opintojakson Moodle-kurssilla myös muita oikeuksia tai valvollisuuksia.

On huomattava, että näissä opettajan ohjeissa opettajan ja assistentin roolit ovat lähes päinvastaiset kuin Aalto-yliopiston. Roolit eivät välttämättä vastaa henkilöiden työnimikkeitä: Sama henkilö voisi luennoida kahta opintojaksoa, mutta toimia STACKissa toisen opettajana ja toisen assistenttina.

## 5.2 Ohjeiden tavoitteet

Interaktiivisten ja automaattisesti tarkastettavien matematiikan tehtävien käyttö voisi tuoda opetukseen merkittävää lisäarvoa. Useimmilla opettajilla ei kuitenkaan tällä hetkellä ole riittävästi tietoa ja taitoa sellaisten luomiseen. STACKin käytön yleistymistä voidaan vauhdittaa tarjoamalla opettajille:

- 1) järjestelmän tarpeita vastaava, keskitetysti ylläpidetty palvelinympäristö
- 2) tukea tehtävien luomiseen ja
- 3) tietoa STACKin käytön eduista.

Opettajan ohjeet pyrkivät täyttämään nämä tarpeet: Ne auttavat opettajia ymmärtämään, millainen työkalu STACK on ja millä tavoin sitä voidaan hyödyntää opetuksessa, ja

antavat konkreettisia ohjeita STACKin ja Moodlen käyttöön. Ohjeisiin kuuluu myös ylläpitäjille suunnattuja STACKin asennukseen ja ylläpitoon liittyviä hyviä käytäntöjä.

Ohjeiden kirjoittamisen ja jakelun tavoitteena on hyvien käytäntöjen vakiinnuttaminen. Tavoite on, ettei STACK tulevaisuudessa ole vain parin aktiivisimman opettajan harrastus, vaan työkalu, jota kaikki pystyvät hyödyntämään.

Ohjeissa ei kuitenkaan käsitellä matematiikan tehtävien laatimiseen liittyvää pedagogiikkaa tai didaktiikkaa. Opettajien oletetaan olevan jo sekä matematiikan että opetuksen ammattilaisia, eikä ohjeiden tarkoitus ole auttaa heitä hankkimaan kyseisiä taitoja vaan soveltamaan jo osaamiaan asioita uudessa ympäristössä

### 5.3 Ohjeiden toteutus

Opettajan ohjeet julkaistaan wiki-sivuina Tampereen teknillisen yliopiston Matematiikan laitoksen ylläpitämässä wiki-ympäristössä. Ohjeiden sisältämä muotoiltu teksti, linkit, kuvat ja taulukot voidaan helposti esittää wikissä.

Wiki-esitysmuodon toinen etu on sen helppo muokattavuus. Kuka tahansa wikin käyttäjä pystyy korjaamaan havaitsemansa virheen, päivittämään vanhentunutta tietoa ja lisäämään omaa sisältöä. Tarkoituksena ei ole päivittää opettajan ohjeita tulevaisuudessa keskitetysti, vaan kannustaa opettajia kirjaamaan ylös omia hyviksi havaitsemiaan käytäntöjä ja kommentoimaan muiden ehdotuksia tarkoitukseen varatulla Moodle-keskustelualueella.

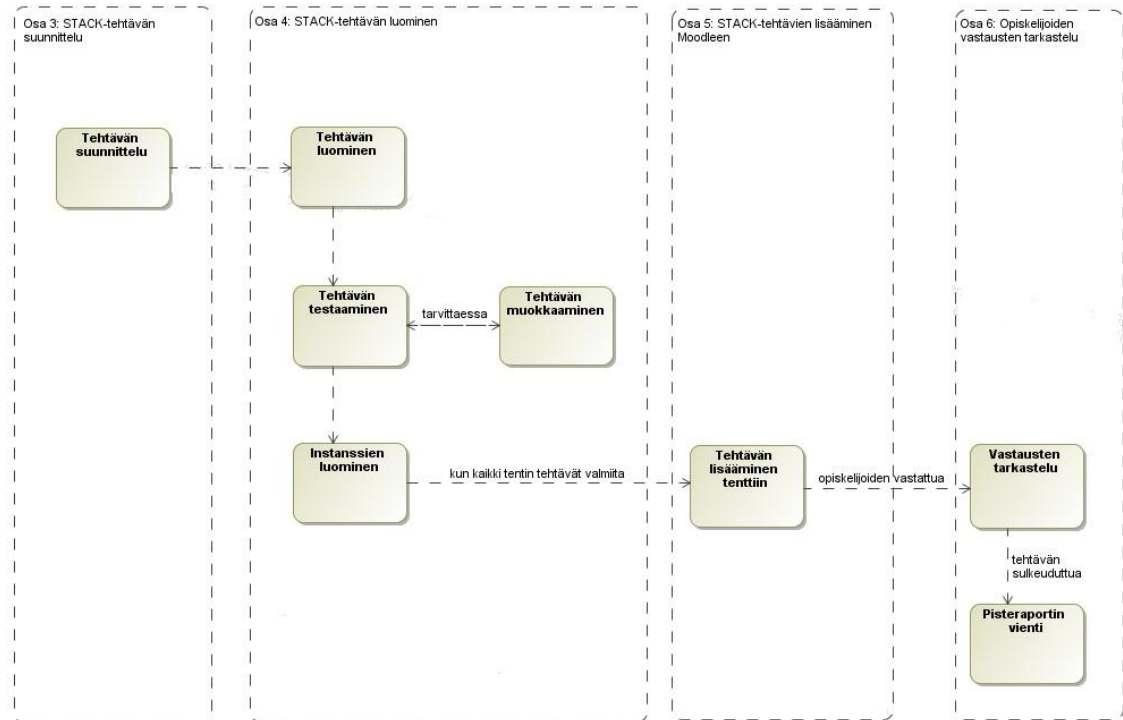
Ohjeiden lisäksi STACKista kiinnostuneille matematiikanopettajille luodaan tunnukset kehityspalvelimella sijaitsevaan Moodleen, johon on luotu esimerkkitehtäviä. Opettajat voivat käyttää näitä tehtäviä malleina omille tehtävilleen. Opettajan ohjeet sisältävät viittauksia esimerkkitehtäviin.

Kehitysympäristöön kuuluu myös foorumi, jossa opettajat voivat keskustella luomistaan tehtävistä. Opettajan ohjeiden ja foorumin sisältöjen ero on se, että opettajan ohjeet kuvailevat yleisiä hyviä käytäntöjä, foorumissa keskustellaan yksittäisistä tehtävistä. Foorumissa opettajat voivat esitellä luomiaan tehtäviä ja pyytää muilta opettajilta apua vaikkapa tehtävien testaamiseen.

### 5.4 Ohjeiden sisältö

Opettajan ohjeet koostuvat kuudesta osasta. Ensimmäinen osa on suunnattu STACKin ylläpitäjälle, joka ei ole matematiikanopettaja vaan tietotekniikan ammattilainen. Toinen osa esittelee opintojakson hallinnan Moodlella ja on suunnattu opettajille ja assistenteille. Sekä kolmas osa, joka käsittelee STACK-tehtävien mahdollisuuksia ja parhaita käytäntöjä.

täntöjä, että käytännön neuvoja STACK-tehtävien luomiseen antava neljäs osa on suunnattu opettajille. Viides osa kertoo STACK-tehtävien lisäämisestä Moodleen ja kuudes opiskelijoiden vastausten tarkastelusta ja Moodlen tuottamien raporttien tulkinnasta. Nämä osat on suunnattu opettajille ja assistenteille. Kuva 5.1 esittää yksittäisen tehtävän elinkaaren opettajan ohjeiden kontekstissa.



**Kuva 5.1.** STACK-tehtävän elinkaari.

Ohjeiden rakenne mukailee STACKin käyttöprosessia: Ylläpitäjä asentaa ensin järjestelmän, minkä jälkeen opettaja luo opintojaksolle Moodle-kurssin, suunnittelee haluamansa interaktiiviset tehtävät, toteuttaa ne ja siirtää ne opiskelijoiden saataville. Opintojakson aikana opettaja seuraa opiskelijoiden vastauksia. Ohjeiden osat voidaan kuitenkin lukea myös itsenäisinä kokonaisuuksina.

#### 5.4.1 Ylläpitäjän työkalut

Ylläpitäjän työkalut -osa on suunnattu henkilöille, jotka vastaavat palvelinten ja tietokonejärjestelmien ylläpidosta. Ylläpitäjille suositellaan kahden eri STACK-version asennusta: yksi tuotanto- ja toinen kehityskäyttöön. Kehityskäyttö tarkoittaa tässä matematiikanopettajien tekemää tehtävien kehitystyötä. Mahdollista itse kehitettyä toiminnallisuutta tai muita teknisiä muutoksia ei ole järkevää testata opettajien käyttämässä kehitysympäristössä, vaan niitä varten suositellaan erillisen kehitysympäristön luomista ohjelmoijille.



Myös yhden STACKin käyttö olisi mahdollista. Sitä ei kuitenkaan suositella, sillä STACK-tehtävät ovat melko haavoittuvia. Luvun 4.2 testien tehtäviä luotaessa havaittiin seuraavat ongelmatilanteet:

- Jos tehtävä on julkaistu opiskelijoille, sen muokkaaminen ja uusien instanssien luominen rikkoo aiemman tehtävän.
- Uuden tehtävän tallentaminen Save-painikkeella (pitäisi valita Save as new) saattaa ylikirjoittaa aiemmin katsellun tehtävän.

Tehtävien vanhojen versioita on mahdollista tarkastella ja palauttaa opettajan käyttöliittymän kautta. Jos tehtävä on vahingossa poistettu kokonaan, se on mahdollista palauttaa tietokannan kautta. Tehtävän palauttaminen ei kuitenkaan palauta siitä luotuja instansseja eikä korjaa rikkoutunutta Moodle-tenttiä. On siis perusteltua erottaa valmiit, opiskelijoille julkaistut tehtävät keskeneräisistä ja viallisista tehtävistä. Samalla kannustetaan opettajia testaamaan tehtävät perinpohjaisesti ennen niiden julkaisua.

Vaihtoehtoisesti olisi mahdollista käyttää kahta STACKia yhden Moodle-ympäristön yhteydessä. Tätäkään ei suositella, sillä on olemassa vaara, että opettajat käyttäisivät tenteissään kehitys-STACKin tehtäviä, jotka voivat muuttua tentin aukioloaikana ja näin rikkoutua.

Ylläpitäjä huolehtii myös käyttöoikeuksien jakamisesta. Suositellaan, että opettajat saavat sekä kehitys-STACKiin että tuotanto-STACKiin opettajan oikeudet. Tämä mahdollistaa tehtävien luomisen ja muokkaamisen molemmissa STACKeissa. Moodle ei oletusarvoisesti sisällä käyttäjäroolia, joka antaisi opettajan luoda tenttejä, muttei uusia STACK-tehtäviä.

Koska STACK2:n kehitys on lopetettu eikä uusien päivitysten julkaisu ole todennäköistä, ylläpitäjälle ei anneta ohjeita STACKin päivittämisestä. Myöskään STACK3:een siirtymistä ei toistaiseksi suositella, sillä sen kehitys on ohjeiden julkaisun aikaan vielä alkuvaiheissa.

Ylläpitäjille annetaan myös TTY:llä käytössä olevaan, luvussa 4.1 kuvattuun asennuspakettiin perustuvat asennusohjeet. Merkittävimmät erot STACKin jakeluversion asennusohjeisiin johtuvat matemaattisten merkintöjen esittämiseen käytetyn JavaScript-kirjaston vaihdosta jsMathista MathJaxiin.

#### **5.4.2 Moodle-opintojakson hallinta**

Moodle-opintojakson hallinta -osa ei juuri esittele Moodlen ominaisuuksia tai käyttöä, vaan opettaja ohjataan hankkimaan tietoa Moodlen dokumentaatiosta tai TTY:n Moodlen ylläpidolta. Opettajalle kerrotaan lyhyesti, kuinka hän voi tilata opintojaksolle Moodle-kurssin ja kuinka opiskelijat voivat liittyä sille.

Moodlen tarjoamia toimintoja käydään läpi niiltä osin kuin ne liittyvät STACKiin. Pääpaino on tenteissä ja niiden käytössä. Opettajalle kerrotaan, millaisilla asetuksilla ja reunaehdoilla STACK-tehtäviä sisältäviä tenttejä voidaan käyttää pisteytettävien tehtävien, kertaustehtävien ja luento-esimerkkien luomiseen. Eri tehtävätyypeille etsitään havainnollistukseksi vastineita luokkamuotoisesta opetuksesta. Kokeiden luomista ei mainita, sillä sitä ei suositella havaittujen teknisten ongelmien ja vilppimahdollisuuksien vuoksi. Opettajille kerrotaan, ettei STACK-tehtäviä sisältävien tenttien kopiointi ole mahdollista.

### 5.4.3 STACK-tehtävän suunnittelu

STACK-tehtävän suunnittelu -osa esittelee STACKin ominaisuuksia ja rajoituksia. Osa on suunniteltu siten, että sitä voidaan käyttää myös STACKin esittelyyn henkilöille, jotka eivät ole aiemmin käyttäneet järjestelmää.

Ensin käydään läpi STACK-tehtävissä tavallisesti esiintyviä asioita: matemaattisten merkintöjen käyttö tehtävänannossa, matemaattiset lausekkeet vastauksina, muuttujien satunnaistaminen, kuvaajat, vektorit ja matriisit ja moniosaisuus. Opettajille kerrotaan, ettei STACKilla ole mahdollista luoda tehtäviä, jotka muuttuvat opiskelijan syötteen perusteella. STACKin kanssa on mahdollista käyttää lisäosia, kuten GeoGebraa-matematiikkaohjelmistoa tai Google Chart -työkaluja, jotka mahdollistavat monipuolisen visualisoinnin ja graafien luomisen. Niitä ei kuitenkaan ole otettu TTY:llä käyttöön, joten niitä ei mainita opettajan ohjeissa.

Tarkastuspuut ovat yksi STACKin suurimmista vahvuuksista, joten niiden toiminta esitellään perusteellisesti. Opettajille kerrotaan ja näytetään esimerkkien avulla, kuinka tarkastuspuilla tarkastetaan ja pisteytetään opiskelijoiden vastauksia. Erityistä huomiota kiinnitetään tapauksiin, joissa sama tarkastuspuu tarkastaa useita vastauskohtia tai samaa vastauskohtaa käsitellään useassa tarkastuspuussa. Opettajille annetaan myös vinkkejä siitä, kuinka tarkastuspuita voidaan käyttää räätälöidyn palautteen antamiseen opiskelijalle. Palautteesta annetut suositukset perustuvat Hattien ja Timperleyn tutkimukseen (Hattie & Timperley 2007).

Opettajia kannustetaan sisällyttämään tehtäviinsä sekä ei-triviaaleja tarkastuksia että mallivastauksia. Aalto-yliopistossa on havaittu, että mallivastauksen luominen on yleensä tehtävän luomisprosessin eniten aikaa vievä osa (Blåfield 2009), mutta opiskelijat myös lukevat niitä ja oppivat niistä (Majander & Rasila 2011). Opettajia kannustetaan myös dokumentoimaan tehtävänsä kehityspalvelimen Moodleen tarkoitusta varten luodulle keskustelualueelle.

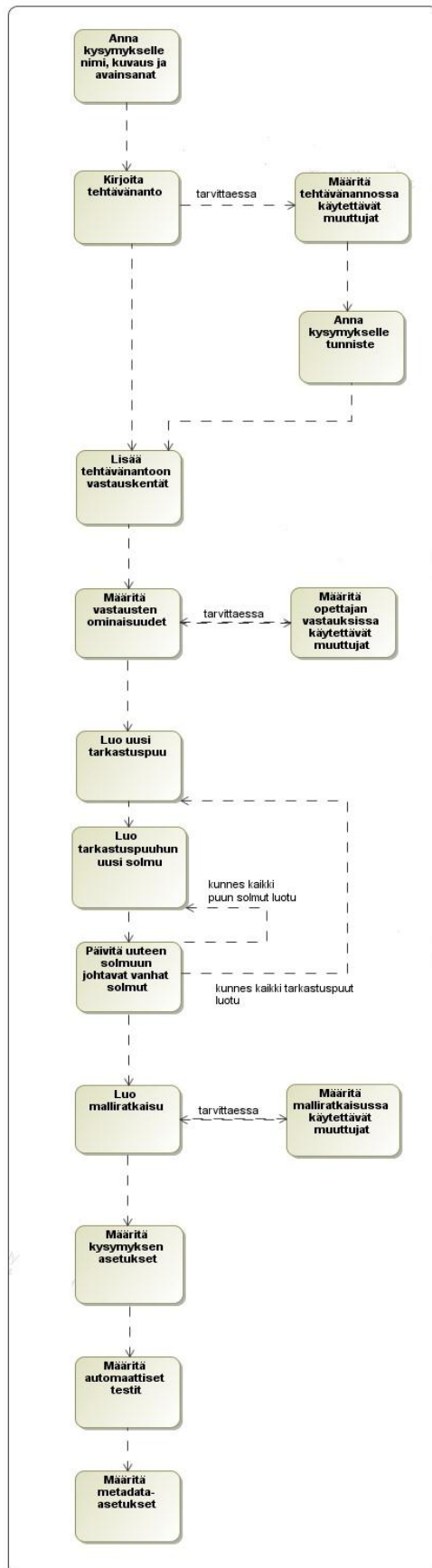
#### 5.4.4 STACK-tehtävän luominen

Kuva 5.2 tarkastelee lähemmin kuvassa 5.1 esiintynyttä tehtävän luomisprosessia. Tehtävän luomisen jälkeen sitä voidaan joutua vielä muokkaamaan. Muokkaamiseen käytetään samaa käyttöliittymää kuin tehtävän luomiseen.

STACK-tehtävän luominen -osa antaa käytännön neuvoja yksittäisen tehtävän luomiseen. Luvussa 3.2.2 esitetyt tehtävän luomisen käyttöliittymän kentät ja niiden mahdolliset arvot käydään läpi. Joidenkin kenttien osalta käsitellään vain ne arvot, joita tavallisimmin käytetään. Esimerkiksi opiskelijan vastausten validointi on syytä jättää aina päälle, jotta tehtävien ratkaisuprosessi pysyy opiskelijan näkökulmasta johdonmukaisena eivätkä opiskelijan väärät oletukset aiheuta pisteiden menettämistä. Pois on jätetty myös joitakin arvoja, joiden on havaittu toimivan epäluotettavasti.

Tehtävän luomisen aikana käytettävät merkintäkielet LaTeX ja XHTML sekä Maxima-funktioiden käyttö esitellään lyhyesti. Niiden käyttöön ei kuitenkaan opasteta syvällisemmin, vaan opettaja ohjataan etsimään lisää tietoa muista lähteistä. Myös STACKin omat merkinnät vastauskentille ja tarkastusten tuloksille esitellään. Tarkastuspuiden suunnittelua käsiteltiin osassa STACK-tehtävän suunnittelu (luku 5.4.3), joten tässä osassa keskitytään niihin liittyviin käytännön valintoihin.

Opettajalle suositellaan tehtävien testaamista sekä STACKin testustoiminnolla että manuaalisesti. Molempiin tapoihin annetaan ohjeet. Instanssien luominen ja tehtävän siirtäminen Moodle-kysymyspankkiin selitetään. Myös tehtävien siirto XML-muodossa STACKista toiseen esitellään.



Kuva 5.2. STACK-tehtävän luomisprosessi.

Tehtävien luomista havainnollistetaan esimerkkitehtävien avulla. Esimerkkitehtävät esitellään vaihe kerrallaan ja niissä tehdyt valinnat perustellaan. Esimerkkitehtävät ovat opettajien saatavilla kehitys-STACKissa.

#### 5.4.5 STACK-tehtävien lisääminen Moodleen

STACK-tehtävien lisääminen Moodleen -osa opastaa opettajaa Moodle-tentin laatimisessa. Tunti on Moodlen nimitys arvosteltavalle tehtäväsarjalle. STACK-tehtäviä sisältäviä tenttejä suositellaan käytettäväksi lisäpiste-, kertaus- ja uuteen asiaan johdattavina tehtävinä, ei kokeina.

Opettajille esitetään suosituksia tentin asetuksista. Koska STACKin opiskelijan käyttöliittymässä on havaittu käytettävyysongelmia, joiden takia opiskelija silloin tällöin vahingossa palauttaa vastauksensa keskeneräisinä, vastauskertojen enimmäismääräksi suositellaan vähintään kolmea, mieluiten rajoittamatonta. Arvioinniksi suositellaan ylintä arviointia, jolloin opiskelijan vastauksista otetaan huomioon eniten pisteitä saanut. On tosin huomattava, että jos tehtävistä ei myönnetä pisteitä samassa suhteessa maksimipistemäärän kanssa<sup>4</sup>, mikään arviointivaihtoehtoista ei anna suoraan oikeita tuloksia.

Kysymysten valinta tenttiin kysymyspankista esitellään. On tärkeää testata kysymykset ennen tentin avaamista. Tuotanto-Moodlen kysymyspankin muokkaamista ei suositella opettajille, sillä auki olevaan tenttiin kuuluvan kysymyksen poistaminen estää opiskelijoita vastaamasta siihen.

#### 5.4.6 Opiskelijoiden vastausten tarkastelu

Opiskelijoiden vastausten tarkastelu -osa opastaa Moodlen arviointityökalujen käyttöön. Opettajalle kerrotaan, kuinka hän voi tarkastella opiskelijoiden koko kurssin pistekertymiä ja tenttien kokonais- ja tehtäväkohtaisia pistemääriä ja tallentaa raportteja ODS-, Excel- ja CSV-muodoissa.

Moodlen kautta tuotettujen pisteraporttien suurin ongelma on niiden yhdistäminen kurssin muihin pisteraportteihin, kuten laskuharjoitustehtäviin ja tenttituloksiin. Tampereen teknillisellä yliopistolla käytetään opiskelijan tunnistamiseen ensisijaisesti opiskelijanumeroa. Moodle sen sijaan ei näytä opiskelijanumeroita ollenkaan, vaan listaa opiskelijat raporteissa pelkän nimen kanssa. Opettaja voi tarvittaessa selvittää Moodlen kautta opiskelijan intranet-tunnuksen, joka toimii yksilöllisenä tunnisteena.

---

<sup>4</sup> Luvussa 4.2.2 kuvatussa testissä opiskelijat suorittivat kolmen STACK-tehtävän sarjan. Jokaisesta oikein suoritetusta tehtävästä opiskelija sai yhden porkkanapisteen. Tehtävien maksimipistemäärät olivat 2,1 ja 6. Oletetaan, että opiskelija suorittaa tehtäväsarjan kahdesti saaden ensimmäisellä yrityksellä tulokset 0/2, 0/1 ja 5/6 ja toisella yrityksellä tulokset 2/2, 1/1 ja 0/6. Tällöin ensimmäisen yrityksen kokonaispistemäärä 5/9 on suurempi kuin toisen yrityksen kokonaispistemäärä 3/9, joten Moodle pitää ensimmäistä yritystä onnistuneempana.

Moodle ei tarjoa työkaluja yksittäisen tehtävän tallentamiseen, mutta niiden tarkastelu on mahdollista. Opettaja näkee opiskelijan saaman tehtävänannon, opiskelijan viimeisimmän vastauksen ja STACKin antaman palautteen. Lisäksi näytetään opiskelijan vastaushistoria. Vastaushistoriaa ei kuitenkaan näytetä viimeisimmän vastauksen lailla vastauskentissä, vaan suoraan STACKin tietokannasta haettuina tekstijonoina. Vastauksesta ei ole suoraan nähtävissä, onko kyseessä validoitu vai tarkastettu syöte. Ohjeissa kerrotaan, kuinka opettaja voi tarvittaessa tulkita vastaushistoriaa vastausten muutosten ja niihin liittyvien arvosanojen perusteella. Vastausyritysten tarkastelu voi tuoda mielenkiintoista tietoa opiskelijoiden kokeilemista ratkaisuksista ja asian ymmärtämisestä. Tämänhetkiset työkalut tarjoavat siihen mahdollisuuden mutta pakottavat opettajan näkemään hieman vaivaa.

Opettajalle kerrotaan myös, kuinka hän voi tarvittaessa pisteittää STACK-tehtäviä manuaalisesti tai muuttaa STACKin antamaa pistemäärää. Näiden toimintojen käytölle ei yleensä pitäisi olla tarvetta. Jos opettaja haluaa arvostella opiskelijoiden vastaukset käsin, tehtävien jakeluun löytyy Moodlesta muitakin keinoja kuin STACK-tehtävät. Moodlen analyysitoiminnon käyttäminen ei STACK-tehtävien yhteydessä ole perusteltua, sillä useimmissa STACK-tehtävissä opiskelijoiden saamat tehtävänannot ja sen vuoksi myös oikeat vastaukset eroavat toisistaan, jolloin vastausten esiintymistiheyksien tilastollinen tarkastelu ei ole mielekästä.

Opiskelijat pystyvät tarkastelemaan tekemiään STACK-tehtäviä ja näkemään niistä saadut pistemäärät. Opettaja ei voi automaattisesti antaa ryhmälle oikeutta tarkastella esimerkiksi tehtävien pistekeskisarvoja.

## 6 JOHTOPÄÄTÖKSET

Matematiikan tehtävien automaattiseen tarkastukseen on kehitetty useita erilaisia järjestelmiä. Niistä kiinnostavimpia ovat matematiikkaohjelmistoihin perustuvat, joita on saatavilla sekä kaupallisina palveluina että avoimen lähdekoodin ohjelmina. Järjestelmän valintaan vaikuttavat käyttökustannuksien lisäksi muun muassa integraatio muihin järjestelmiin, tekninen toimivuus ja käytettävyys sekä tehtäviä luovan opettajan että niitä ratkaisevan opiskelijan näkökulmasta.

Automaattisesti tarkastettavia tehtäviä on aiemmin hyödynnetty Tampereen teknillisellä yliopistolla heikoimpien opiskelijoiden kertaustehtävinä. Tätä varten on kehitetty STACKiin pohjautuva tehtävämoottori MathBridge-oppimisympäristöön. Sen käyttö TTY:n varsinaisilla matematiikan opintojaksoilla ei kuitenkaan ole hyvä vaihtoehto opettajan käyttöliittymän puuttumisen ja rajoitettujen ominaisuuksien vuoksi. Tehtävämoottoria on kuitenkin mahdollista kehittää edelleen ja integroida muihin oppimisympäristöihin.

STACK on matemaattisten aineiden opetuksen apuväline, jolla opettaja voi luoda opiskelijoille automaattisesti tarkastettavia, mukautettua palautetta antavia, tavallisesti käytettyjä matemaattisia merkintöjä hyödyntäviä tehtäviä. Opettajan työ suuntautuu STACKin avulla mekaanisesta tarkastuksesta pedagogisesti mielekkäiden harjoitustehtävien kehittämiseen. Toisaalta yliopisto-opettaja ei tälläkään hetkellä käytä suurta osaa ajastaan tehtävien tarkastamiseen, vaan opiskelijat tarkastavat tehtävänsä harjoitustilaisuuksissa melko itsenäisesti. Yhdessä Moodlen kanssa käytettynä STACK voi vähentää opintojakson hallinnointiin liittyvää työtä. STACK-tehtävien laatiminen vie kuitenkin perinteisiä tehtäviä enemmän aikaa. Opiskelijat suhtautuvat STACK-tehtäviin positiivisesti erityisesti silloin, kun he saavat järjestelmältä ohjaavaa palautetta ja malliratkaisuja.

STACK ei kerro opettajalle, mitä tai miten tulisi opettaa. STACKin menestyksessä käyttö perustuu täysin tehtävät laativan opettajan aihealueen ja pedagogiikan tuntemukseen. Opettajan on myös tunnettava STACKin mahdollisuudet ja työtavat. Opettajan vastuulle jää myös opiskelijoiden motivoiminen, sillä pelkkä STACKin olemassaolo ei saa opiskelijoita laskemaan harjoitustehtäviä aivan kuten pelkkä tekstikirjan olemassaolo ei saa opiskelijoita lukemaan. STACKin käyttöönotto ei olekaan avaimet käteen - ratkaisu, vaan jokaisen sitä käyttävän opettajan on sitouduttava käyttämään sitä parhaalla mahdollisella tavalla.

STACK2:n käyttöä hankaloittavat epäintuitiiviset käyttöliittymät, moninaiset syntaksit ja ajonaikaiset tekniset ongelmat. Useiden ohjelmistoriippuvuuksiensa takia STACK2 on työläs ylläpidettävä, ja koska sen kehitys on jo loppunut, kehittäjät eivät todennäköisesti tarjoa siihen enää uusia päivityksiä. STACKista saadaan enemmän hyötyä, jos sitä mukautetaan ja kehitetään vastaamaan käyttävän oppilaitoksen tarpeita. Lisäksi STACKin ylläpito vaatii jonkin verran teknistä osaamista ja resursseja. STACK soveltuu siis parhaiten oppilaitoksille, joilla on jo tietoa ja kokemusta ohjelmistojen kehittämisestä. Avaimet käteen -ratkaisu se ei ole.

Pedagogisesti STACK soveltuu monenlaisten tehtävien luomiseen. Opiskelijoille voidaan helposti tarjota suuri määrä keskenään hieman erilaisia kohtuullisen yksinkertaisia laskutehtäviä aiemmin opittujen asioiden kertaamiseen tai laskurutiinin vahvistamiseen. Toisaalta STACKilla voidaan luoda myös pitkiä, monivaiheisia tehtäviä, jotka johdattavat opiskelijat uuteen aiheeseen tai syventävät heidän tietämystään aiemmin käsitellystä asiasta. Tehtäviä ja mallivastauksia voidaan rikastaa kuvaajilla. Teknisten ongelmien vuoksi STACKia ei kuitenkaan pitäisi käyttää tiukasti arvosteltavien tehtävien luomiseen. Vaikka STACK tarjoaa paljon yksityiskohtaista pisteytystoiminnallisuutta, käyttöliittymän sekavuuden, syntaksin tuntemattomuuden ja ajonaikaisten virheiden vuoksi tehtävistä saamat pistemäärät eivät täysin vastaa opiskelijoiden todellista osaamista. Jos STACK-tehtäviä käytetään osana opintojakson lopullista arvostelua, esimerkiksi pakollisina harjoituksina tai ns. porkkanapistetehtävinä, hyväksytyin suorituksen rajana ei voida pitää täydellistä suoritusta, ja opiskelijoille on annettava mahdollisuus suorittaa tehtäväsarja useammin kuin kerran.

Tämän diplomityön osana toteutetut opettajan ohjeet antavat konkreettisia ohjeita STACKin käyttöön ja auttavat opettajia ymmärtämään, millaisten tehtävien luomiseen se soveltuu. Ohjeisiin on kirjattu järjestelmän ylläpitoon, Moodlen käyttöön, tehtävien suunnitteluun ja toteuttamiseen ja opiskelijoiden vastausten tarkasteluun ja tulkintaan liittyviä hyviä käytäntöjä, jotka pyritään vakiinnuttamaan. Hyviä käytäntöjä noudattamalla järjestelmän ja tehtävien ylläpito helpottuu ja teknisten ja käyttöliittymästä johtuvien ongelmien määrä vähenee. Kaikkien STACK2:ta käyttävien on syytä tutustua opettajan ohjeisiin.

Opettajan ohjeita ei ole tämän diplomityön puitteissa testattu, mikä vaikeuttaa niiden laadun arvioimista. Opettajan ohjeisiin olisi syytä lisätä luku, joka käsittelee tehtävien ratkaisemista opiskelijan näkökulmasta, sillä opettaja joutuu todennäköisesti ohjeistamaan opintojaksojensa osallistujia STACKin käytöstä.

Pelkällä käyttäjien ohjeistuksella ei ole mahdollista ratkaista kaikkia järjestelmän ongelmia. Ennen STACKin käyttöönottoa on syytä selvittää, onko STACK automaattisesti tarkastettavia matematiikan tehtäviä luovista järjestelmistä paras haluttuun tarkoituk-



seen, mikä STACKin versioista soveltuu siihen parhaiten ja mitä muutoksia STACKiin halutaan itse tehdä ennen sen käyttöönottoa.

STACKin menestyksellä käyttöönotto edellyttää parannuksia erityisesti opiskelijan käyttöliittymään. Testeissä havaituista ongelmista suurin osa johtui siitä, etteivät opiskelijat ymmärtäneet järjestelmän toimintoja tai syntaksia. Jos STACK otetaan käyttöön matematiikan peruskursseilla, opiskelijoiden määrä on niin suuri, ettei ole järkevää erikseen opettaa kaikille STACKin käyttöä. Järjestelmä on siis saatava toimimaan intuitiivisesti. Ensivuksi auttaisivat jo selkeämmät ilmoitukset tehtävän tarkastuksen vaiheesta ja opiskelijan seuraavasta toimenpiteestä.

STACKin käyttöön on varattava huomattavia resursseja. Opettajia on motivoitava ja koulutettava STACK-tehtävien luomiseen ja tehtävien tasoa ja oikeellisuutta on ainakin aluksi valvottava keskitetysti. Lisäksi järjestelmäylläpitäjien on tarjottava STACKille käyttötukea. Vaikka STACK on ilmainen, sen käyttö ei tule halvaksi. Paras hintalaatusuhde tehtäville saataisiin, jos ne suunniteltaisiin ja toteutettaisiin MathBridgen kaltaisena projektina yhdessä muiden yliopistojen kanssa. Tällöin opettajien käyttöön voitaisiin tarjota suurempi määrä valmiita tehtäviä, mikä laskisi kynnystä automaattisten tehtävien käyttöönottoon.

Toinen harkitsemisen arvoinen vaihtoehto on ottaa STACKista käyttöön vain opettajan käyttöliittymä ja siirtää sillä luodut tehtävät XML-tiedostoina toiseen oppimisympäristöön, jossa ne arvostellaan toisella tehtävämootorilla. Tällöin saadaan käyttöön STACKin parhaat puolet – tehtävien ja tarkastuspuiden toiminnallisuus kätevässä esitysmuodossa ja riittävän hyvin toimiva opettajan käyttöliittymä – ja voidaan itse luoda parempi opiskelijan käyttöliittymä. Tämä vaihtoehto ei kuitenkaan ole koko STACKin käyttöönottoa kustannustehokkaampi.

STACK on mielenkiintoinen järjestelmä, jossa on paljon potentiaalia. Tällä hetkellä STACK soveltuu kuitenkin vasta yksittäisten, erittäin kiinnostuneiden opettajien leikkikaluksi tai alkavaksi ohjelmistokehitysprojektiksi. Koko yliopiston laajuiseen käyttöön sitä ei voi nykyisessä tilassaan suositella, mutta tilanne voi muuttua nopeastikin STACK3:n valmistuessa.

## LÄHTEET

- Allan, B. 2007. Blended Learning. Trowbridge, Cromwell Press Ltd. 228 s.
- Blåfield, L. 2009. Matematiikan verkko-opetus osana perusopetuksen kehittämistä Teknillisessä korkeakoulussa. Pro gradu -tutkielma. Helsinki. Helsingin yliopisto. 49 s.
- Bokhove, C. & Drijvers, P. 2010. Digital Tools for Algebra Education: Criteria and Evaluation. *International Journal of Computers for Mathematical Learning* [verkkolehti]. 15, 1, pp. 45-62. 30.4.2010 [viitattu 20.3.2012]. Saatavissa: <http://www.springerlink.com/content/102910/>.
- Deutsches Forschungszentrum für Künstliche Intelligenz & Saarland University. ActiveMath Software [WWW]. 2007 [viitattu 2.4.2012]. Saatavissa: <http://www.activemath.org/Software>.
- Embacher, F., Tonti, F. & Spreitzer, C. 2011. Report on User Feedback Collection Process and Results. 76 s.
- Freudenthal Institute. WisWeb [WWW]. 2004 [viitattu 3.4.2012]. Saatavissa: <http://www.fi.uu.nl/wisweb/en/>.
- Harjula, M. 2008. Mathematics Exercise System with Automatic Assessment. Diplomityö. Helsinki. Teknillinen korkeakoulu, Automaatio- ja systeemitekniikan tutkinto-ohjelma. 94 s.
- Hattie, J. & Timperley, H. 2007. The Power of Feedback. *Review of Educational Research* 77, 1, pp. 81-112.
- Hrastinski, S. 2008. Asynchronous and Synchronous E-Learning. *Educause Quarterly* [verkkolehti]. 31, 4. [viitattu 17.3.2012]. Saatavissa: <http://www.educause.edu>.
- Huikkola, M., Silius, K. & Pohjolainen, S. 2008. Clustering and achievement of engineering students based on their attitudes, orientations, motivations and intentions. *WSEAS Transactions on Advances in Engineering Education* 5, 5, pp. 342-354.
- Hunt, T. Open Protocol for Accessing Question Engines [WWW]. 26.9.2011 [viitattu 23.3.2012]. Saatavissa: [http://docs.moodle.org/dev/Open\\_protocol\\_for\\_accessing\\_question\\_engines](http://docs.moodle.org/dev/Open_protocol_for_accessing_question_engines).
- Itkonen-Isakov, T-M. Mikä oppimisessa sulautuu? [WWW]. 18.11.2009 [viitattu 20.3.2012]. Saatavissa:

[http://www.virtuaalikorkeakoulupaivat.fi/palvelut/drupal/sites/default/files/esitykset/19C\\_itkonen.pdf](http://www.virtuaalikorkeakoulupaivat.fi/palvelut/drupal/sites/default/files/esitykset/19C_itkonen.pdf).

Keränen, V. & Penttinen, J. 2007. Verkko-oppimateriaalin tuottajan opas. Porvoo, WSOY. 293 s.

Korhonen, V. & Pantzar, E. 2004. Verkko-opetuksen ja vuorovaikutuksen erityispiirteitä tunnistamassa. In: Korhonen, V. (toim.). Verkko-opetus ja yliopistopedagogiikka. Tampere, Tampere University Press. pp. 17-45.

Majander, H. & Rasila, A. 2011. Experiences of continuous formative assessment in engineering mathematics. In Silfverberg, H., Joutsenlahti, J. (toim.). Tutkimus suuntaamassa 2010-luvun matemaattisten aineiden opetusta. Tampereen yliopistopaino Oy - Juvenes Print. pp. 197-214.

MapleSoft. Maple T.A. [WWW]. 2012 [viitattu 3.4.2012]. Saatavissa: <http://maplesoft.com/products/mapleta/index.aspx>.

Matematiikkajumppa [WWW]. 13.10.2011 [viitattu 19.3.2012]. Saatavissa: <http://hlab.ee.tut.fi/piiri/groups/matematiikkajumppa>.

MathBridge [WWW]. 2010 [viitattu 23.3.2012]. Saatavissa: <http://www.math-bridge.org/>.

Mathematical Association of America. WeBWorK [WWW]. [viitattu 3.4.2012]. Saatavissa: <http://webwork.maa.org/>.

Moodle. [WWW]. [viitattu 3.5.2012]. Saatavissa: <http://www.moodle.org>.

Moodle. Question Types [WWW]. [viitattu 4.5.2012]. Saatavissa: [http://docs.moodle.org/19/en/Question\\_types](http://docs.moodle.org/19/en/Question_types).

Moodle. Standard Moodle Packages [WWW]. [viitattu 8.4.2012]. Saatavissa: <http://download.moodle.org/>.

Nykänen, O. & Pohjolainen, S. 1998. Using General Purpose Computer Algebra System Maple in WWW-Based Open Learning Environments. In: Multisilta, J. & Niemi, J.-P. (toim.). Proceedings of the LeTTeT'98 and MaTILDA'98 Joint Conference. Pori. pp. 73-80.

Nykänen, O. 2000. A Framework for Generating Non-trivial Interactive Mathematical Exercises in the Web: Dynamic Exercises. In: Bourdeau, J. & Heller, R. (toim.). ED-

MEDIA 2000, World Conference on Educational Multimedia, Hypermedia & Telecommunications. Montreal, Kanada. pp. 1438-1439.

Ojala, T. 2004. Riittääkö aika, riittävätkö rahat? – Tutkimus verkko-opetuksen työmääristä ja kustannuksista. In: Kähkönen, E. (toim.). Verkko-oppimisen vakiintuessa. Joensuu, Joensuun yliopisto. pp. 219-257.

Opetushallitus. Verkko-oppimateriaalit [WWW]. 13.3.2012 [viitattu 15.3.2012]. Saatavissa: [http://www.edu.fi/verkko\\_oppimateriaalit](http://www.edu.fi/verkko_oppimateriaalit).

Pantzar, E. 2004. Oppimisympäristö verkkona – verkko oppimisympäristönä. In: Korhonen, V. (toim.). Verkko-opetus ja yliopistopedagogiikka. Tampere, Tampere University Press. pp. 49-68.

Pearson Education. MathXL [WWW]. 2012 [viitattu 3.4.2012]. Saatavissa: <http://www.mathxl.com>.

Pirttimäki, S. 2004. Yliopisto-opettajien verkko-opetuksessa kohtaamat pedagogiset haasteet. In: Kähkönen, E. (toim.). Verkko-oppimisen vakiintuessa. Joensuu, Joensuun yliopisto. pp. 219-257.

Pohjolainen, S., Miilumäki, T., Kangas, J., Nykänen, O., Mäkelä, T. & Sarikka, H. TUT Final Internal Math-Bridge Project Report. 16.3.2012. Tampereen teknillinen yliopisto. Julkaisematon luonnos.

Pohjolainen, S., Raassina, H., Silius, K., Huikkola, M. & Turunen, E. 2006. TTY:n insinöörimatematiikan opiskelijoiden asenteet, taidot ja opetuksen kehittäminen. Tampere, Tampereen teknillinen yliopisto, Matematiikan laitos, Tutkimusraportti 84. 127 s.

Rasila, A., Harjula, M. & Zenger, K. 2007. Automatic Assessment of Mathematics Exercises: Experiences and Future Prospects. In: Yanar, A. & Saarela-Kivimäki, K. (toim.). ReflekTori 2007 Symposium of Engineering Education. Helsinki, Teknillinen korkeakoulu. pp. 70-80.

Rasila, A., Havola, L., Majander, H. & Malinen, J. 2010. Automatic Automatic assessment in engineering mathematics: evaluation of the impact. In: Myller, E. (toim.), ReflekTori 2010 Symposium of Engineering Education. Aalto University School of Science and Technology. pp. 37-45.

Ruokokoski, J. 2009. Automatic Assessment in University-Level Mathematics. Diplomityö. Helsinki. Teknillinen korkeakoulu, Teknillisen fysiikan koulutusohjelma. 77 s.

Sangwin, C. 2003. Assessing Mathematics Automatically Using Computer Algebra and the Internet. *Teaching Mathematics Applications* 23, 1, pp. 1-14.

Sangwin, C. 2007. STACK: Making Many Fine Judgements Rapidly. *CAME*. Julkaisematon artikkeli. 15 s.

Sangwin, C. 2010. Who Uses STACK? A Report on the Use of the STACK CAA System. Maths, Stats and OR Network, School of Mathematics, University of Birmingham. 15 s.

Sangwin, C. STACK Version 3.0 [WWW]. 3.3.2012 [viitattu 23.3.2012]. Saatavissa: <http://moodle.org/mod/forum/discuss.php?d=197507>.

Sangwin, C. & Grove, M. 2006. STACK: Addressing the Needs of the "Neglected Learners". In: *Proceedings of the WebAlt Conference*. Eindhoven, Alankomaat. pp. 81-95.

Sosa, G., Berger, D., Saw, A. & Mary, J. 2011. Effectiveness of Computer-Assisted Instruction in Statistics: A Meta-Analysis. *Review of Educational Research* 81, 1, pp. 97-128.

StackWiki [WWW]. 10.5.2011 [viitattu 28.3.2012]. Saatavissa: [http://www.stack.bham.ac.uk/wiki/index.php/Main\\_Page](http://www.stack.bham.ac.uk/wiki/index.php/Main_Page).

StackWiki. Development Track [WWW]. 10.5.2011 [viitattu 23.3.2012]. Saatavissa: [http://www.stack.bham.ac.uk/wiki/index.php/Development\\_track](http://www.stack.bham.ac.uk/wiki/index.php/Development_track).

Tampereen teknillinen yliopisto. 2009. Koulutuksen kehittämisohjelma vuosille 2009-2015. Tampere, Tampereen teknillinen yliopisto, Yliopistopalvelut, Raportti 4. 25 s.

University of York. AiM Assessment in Mathematics [WWW]. [viitattu 3.4.2012]. Saatavissa: <http://maths.york.ac.uk/yorkmoodle/course/view.php?id=67>.

Xiao, Gang. Interactive Mathematics on the Internet [WWW]. [viitattu 3.4.2012]. Saatavissa: <http://wims.unice.fr/>.