Institutionen för beteendevetenskap och lärande
LiU-PEK-R-267
Licentiatavhandling

Grundskoleelevers design i lärande

En studie om lärprocesser i programmering

Maria Sparf

Licentiatavhandling

Grundskoleelevers design i lärande

En studie om lärprocesser i programmering

Maria Sparf

Institutionen för beteendevetenskap och lärande

Linköpings universitet

LiU-PEK-R-267

November 2020

LINKOPINGS UNIVERSITET
Institutionen för beteendevetenskap och lärande
LiU-PEK-R-267

ISBN: 978-91-7929-725-1

Detta verk är licensierat under en Creative Commons Erkännande-IckeKommersiell 4.0 Internationell Licens.

<https://creativecommons.org/licenses/by-nc/4.0/deed.sv>

Linköpings universitet
Institutionen för beteendevetenskap och lärande
SE-581 83 Linköping, Sweden
Tel 013-28 10 00

Tryck: Linköpings universitet, LiU-Tryck 2021

Till alla elever jag mött!

”I am magic!”

Anna, årskurs 5

INNEHÅLL

GRUNDSKOLEELEVERS DESIGN I LÄRANDE	1
ABSTRACT	1
SVENSK SAMMANFATTNING	3
INGÅENDE ARTIKLAR	5
FÖRKORTNINGAR	6
TACK	7
1. INTRODUKTION	9
1.1 Programmering i en digitaliserad värld	9
1.1.1 Elevperspektiv	10
1.1.2 Science center som lärmiljö	11
1.1.3 Designorienterad teori	11
1.2 Syfte och frågeställning	12
1.3 Studiens disposition	12
2. BAKGRUND	14
2.1 Digitalisering och programmering i svensk skola	14
2.1.1 Digital kompetens.....	14
2.1.2 Digitalisering av skolan	15
2.1.3 Programmering in på schemat.....	15
2.1.4 Undervisning för nybörjare i skolmiljö.....	18
2.1.5 Förutsättningar för att undervisa i programmering	19
2.1.6 Datalogiskt tänkande.....	20
2.2 Science centers	21
2.2.1 Föreningen Svenska Science Centers.....	22
2.2.2 Lärmiljö i ett spänningsfält	22
2.2.3 Programmeringsverktyg på SC	25
3. TIDIGARE FORSKNING	27

3.1 Hård och mjuk programmering	27
3.2 Visuell kod eller textbaserad kod.....	28
3.3 Datalogiskt tänkande och programmering.....	31
3.4 Engagemang för att lära programmering.....	33
3.5 Syntes.....	36
4. TEORETISKA PERSPEKTIV.....	38
4.1 Designorienterad teori.....	38
4.1.1 Informellt och formellt lärande	38
4.1.2 Design för lärande.....	39
4.1.3 Learning Design Sequences	41
4.2 Engagemang	43
4.2.1 Definition av engagemang.....	44
4.2.2 Engagemang och lärande.....	46
5. METOD	47
5.1 Studiens genomförande.....	47
5.1.1 Urval.....	47
5.1.2 Datainsamlingen	49
5.2 Analys	51
5.2.1 Tematisk analys	51
5.2.2 Analys artikel 1	52
5.2.3 Analys artikel 2.....	53
5.3 Metoddiskussion, etik och kvalitet	53
5.3.1 Etiska överväganden	53
5.3.2 Studiens kvalitet.....	55
6. RESULTAT.....	58
6.1 Artikel 1:	58
6.1.1 Förutsättningar för lärande i programmering på SC.....	58
6.1.2 Transformationsprocess leder till olika tillvägagångssätt.....	60
6.1.3 Sammanfattning av resultat i artikel 1	64
6.2 Artikel 2:	65
6.2.1 Förutsättningar för engagemang	65
6.2.2 Exempel för att belysa olika typer av engagemang.....	66
6.2.3 Sammanfattning av resultat i artikel 2	67

7. DISKUSSION	68
7.1 Lärprocesser i programmering.....	68
7.1.1 Elever som designers.....	68
7.1.2 Iscensättning, möjligheter och dilemman.....	71
7.1.3 Tillvägagångssätt och engagemang	73
7.2 Design för lärande i studien	75
7.3 Studiens bidrag.....	77
7.4 Vidare forskning.....	78
REFERENSER	79
APPENDIX.....	88
TIDIGARE UTGIVNA RAPPORTER	91

ABSTRACT

The aim of this study is to contribute to the knowledge about how pupils design their learning in programming. It is mainly the learning process, how pupils deal with problems in programming and how they become, are and remain engaged in the tasks, which is of interest. Programming can be understood in many ways; coding, a digital competence, creativity, or ways to solve problems. The digitalisation of society has also evoked a need to learn programming from an early age in compulsory school. In this study, programming is seen as a part of the digital competence that all pupils should have the opportunity to develop, which is a common thread that runs throughout compulsory school.

The study was conducted during programming lessons at three science centres in Sweden. The centres had previous experience in teaching programming. This was used at the time of the study by schools that in this way could offer pupils to try programming even before it became part of the compulsory teaching. The lessons were adapted for novices in programming and were conducted as part of the regular school day for pupils in grades 1-8.

The theoretical framework is based on design-oriented theory with a focus on how settings and design for learning includes both opportunities and dilemmas for learning. It provides a basis for the analysis of pupils' approaches when learning programming as well as how different types of engagement relates to their design in learning. The results are presented in two articles, which contribute with different aspects of learning. Together the articles provide a picture of pupils' learning design within programming in compulsory school.

The first article highlights knowledge of five different approaches that pupils used to solve assignments using programming. The qualitatively different ways that pupils used during the observed lessons were *mathematically*, *trial and error*, *step-by-step*, *routine* as well as *aesthetically*. Each of these approaches allows pupils to use and practice different abilities that are important for programming. The abilities are compared to, but not equal to computational thinking (CT), which (in its

turn) is linked to competencies that are important for an active participation in a digital society.

The second article contributes to the understanding of how *behavioural*, *emotional*, and *cognitive* engagement can be identified when pupils are learning programming. To understand how the different types of engagement are individually important, yet intertwined and influencing each other, is keen knowledge. The results show how different types of engagement become visible during programming lessons. Furthermore, it is discussed how the pupils' identified engagement can be related to how their learning process is designed.

In the study, taken as a whole, the results of the two articles show how pupils become designers in their programming learning process. The pupils designed their learning throughout their learning process regarding to the settings, to the approach they used and in the way they became engaged.

Keywords: programming, approach, engagement, design for learning, computational thinking science centre

SVENSK SAMMANFATTNING

Studiens övergripande syfte är att fördjupa kunskapen om hur elever designar sitt lärande i programmering. Det är främst elevernas lärandeprocess, hur de tar sig an problem inom programmering och hur de blir, är och förblir engagerade i uppgifterna, som är av intresse. Programmering kan förstås på många olika sätt, kodning, en digital kompetens, kreativitet eller sätt att lösa problem. Digitaliseringen i samhället har även aktualiserat behovet av att lära sig programmering redan från tidig ålder i grundskolan. I denna studie ses programmering som en del av den digitala kompetens alla elever ska ha möjlighet att utveckla och som finns med som en röd tråd genom hela grundskolan.

Studien genomfördes under programmeringslektioner på tre science centers i Sverige. Science center har lång erfarenhet av att undervisa i programmering för barn och unga. Detta nyttjades vid tidpunkten för studien av skolor som på det sättet kunde erbjuda eleverna att prova på programmering redan innan det blev en del av den obligatoriska undervisningen. Lektionerna var anpassade för nybörjare i programmering och genomfördes som en del av skoldagen för elever i årskurs 1-8.

Det teoretiska ramverket har utgångspunkt i designorienterad teori, med fokus på hur iscensättning och design för lärande, som omfattar både möjligheter och dilemman för lärande. Ramverket ger en grund för hur elevers läraprocess och engagemang för att lära sig programmering kan analyseras.

Resultaten redovisas i form av två artiklar som tillsammans ge en bild av design för och i lärande under programmeringslektioner. Den första artikeln bidrar med kunskap om fem olika tillvägagångssätt (i artikel 1 på engelska, approach) som elever använde för att lösa uppgifter med hjälp av programmering. De kvalitativt olika sätt som eleverna använde under de observerade lektionerna var *matematiskt*, *fel- och försök igen*, *steg-för-steg*, *rutin* samt *estetiskt*. Var och ett av dessa tillvägagångssätt gav eleverna möjlighet att använda och träna olika förmågor som är viktiga för att kunna programmera. Förmågorna jämförs, men likställs inte med datalogiskt tänkande (CT) vilket kan anses vara knutet till kompetenser som är viktiga för att aktivt kunna delta i ett digitalt samhälle.

Den andra artikelns kunskapsbidrag är att förstå hur *beteendemässigt*, *emotionellt* och *kognitivt* engagemang kan identifieras när elever programmerar. Att förstå hur de olika typerna av engagemang är viktiga var och en för sig, samtidigt som de är sammanflätade och påverkar varandra, är angelägen kunskap. Resultaten visar hur olika typer av engagemang blir synliga under programmeringslektioner. Vidare diskuteras hur elevernas identifierade engagemang kan relateras till hur deras lärprocess designas.

I den sammanlagda studien visar resultaten från de båda artiklarna på hur elever blev designers för och i sitt lärande i programmering. Eleverna designade sitt lärande genom hela lärprocessen med hänseende till iscensättningen, vilka tillvägagångssätt de använde och hur de hade möjlighet att vara engagerade på olika sätt.

Nyckelord: programmering, tillvägagångssätt, engagemang, design för lärande, datalogiskt tänkande, science center

INGÅENDE ARTIKLAR

I. Design for learning programming

Approaches taken by novice learners

(Sparf, M., Löfgren, H., & Kreitz-Sandberg, S. inskickad)

II. “I am magic!”

Pupils’ engagement when designing in learning programming

(Sparf, M., inskickad)

Maria Sparf står som huvudförfattare till den första artikeln. Sparf står för all datainsamling, litteraturgenomgång kring tidigare forskning och teoretiskt ramverk samt textproduktion till artikeln. Författarna står tillsammans för analys och diskussion av resultatet.

Maria Sparf är författare till artikel nr 2.

FÖRKORTNINGAR

CT	Computational thinking vilket på svenska översätts med datalogiskt tänkande
LDS	Learning design sequences, modell över lärssekvens
IKT	Informations- och kommunikationsteknik
SC	Science centre vilket på svenska skrivs ut science center

TACK

Jag vill börja med att rikta ett stort och varmt tack till mina handledare, som trots att ni båda under min studietid gick vidare till nya uppdrag behöll mig som doktorand! Susanne Kreitz-Sandberg, huvudhandledare, tack för alla givande, vägledande samtal, både hemma på din kökssoffa likaväl som via zoom från Australien. Håkan Löfgren, biträdande handledare, tack för alla kloka ord och tankar. Utan er handfasta hjälp, även med all text vi strukit, ingen färdig studie. Ingen av oss kunde speciellt mycket om programmering när vi började- men nu vet vi desto mer! Åtminstone om hur grundskoleelever gör när de designar sitt lärande i programmering!

Livet består inte bara av arbete och studier. Hemma har jag under min studietid haft fullt stöd och förståelse för hur studier kan engagera känslomässigt till tårar av både frustration och eufori, men även kognitivt för att inte ge upp och bli klar i tid till deadline med detta arbete. Håkan, långa promenader eller att få spela musik tillsammans är bästa sättet för mig att koppla av och det vet du så väl! Tack för att du alltid tror på mig och fortsätter att uppmuntra mig till att göra det jag trivs med. Ludvig, Amanda och Alisia, tack för att ni finns och förgyller livet på så många vis.

Tack också till alla doktorander som jag träffat i olika kurser. Det är en speciell situation att vara doktorand och ni har fått mig att känna att jag inte är ensam i detta. Att få diskutera tillsammans och läsa era texter har gjort att jag har haft ett sammanhang att höra till. Andreas Larsson, du förtjänar ett särskilt tack. Att mötet den där första dagen på vår magisterutbildning när du såg till att vi hamnade i samma studiegrupp skulle bli så betydelsefullt för mina fortsatta studier det kunde jag aldrig tänka mig. Tack för alla kommentarer på mina texter genom åren, givande diskussioner och uppmuntrande meddelanden. Nu får du köpa mig en keps, hatten får vänta ett tag till!

Tack till Per Gyberg som läste manus vid mitt deltidseminarium och fick mig att inse allt inte är lika viktigt. Tack Susanne Engström som läste manus vid mitt slutseminarium och gav många värdefulla kommentarer som hjälpte mig i slutsputten av mitt arbete. Tack även till övriga personer i betygskommittén, Susanne Kjällander som var opponent, Jonas Hallström som var examinator samt Linda Mannila. Ni hjälpte mig alla att göra mitt arbete bättre.

Att få denna möjlighet att läsa vidare och grotta ner mig i tankar om vad lärande handlar om har varit en ynnest. Därför vill jag även ge ett särskilt tack till min dåvarande chef Anna Karlefjärd som trodde på mig och gav mig möjligheten att bli en av Jönköpings kommuns doktorander. Tack också till utbildningsförvaltningen i Jönköping som under fyra år lät mig ägna halva min arbetstid till studier, vilket har varit en stor förmån och en förutsättning för att detta skulle bli möjligt.

Kära kollegor på Upptech science center och på utvecklings- och kvalitetsenheten, ni har varit inspirationskällor och påhejare, ni saknas mig nu när jag slutat, men vi ses igen i nya sammanhang!

Kaxholmen, hösten 2020

1. INTRODUCTION

”Det här är så coolt, jag hade ingen aning om det funkade så.”

Anna, elev i årskurs 5, vid ett besök på Science center

Anna får låna sin röst för att inleda min licentiatstudie då jag själv många gånger har hört liknande kommentarer under mina verksamma år som pedagog på ett science center (SC). Anna är engagerad i det hon får uppleva när hon ska programmera för första gången. Svenska Science Centers¹ beskriver på sin hemsida att ett besök på ett SC bland annat leder till ökat engagemang för naturvetenskap och teknik både i skolarbete och på ledig tid. Anna är en av de ungdomar jag mötte på ett SC under mitt arbete med datainsamlingen till denna studie, så visst stämmer påståendet på hemsidan, åtminstone för vissa elever. Det leder fram till att i denna studie är det elevernas läroprocess, hur de tar sig an problem inom programmering och hur de blir, är och förblir engagerade i uppgifterna, som är av intresse.

1.1 Programmering i en digitaliserad värld

Digitaliseringen pågår i hela världen och digital teknik spelar en stor roll i flera aspekter av ett barns liv. Digital kompetens kan därför anses vara en nödvändig kunskap för alla barn och ungdomar inför framtiden (Fransson et al., 2018; Siddiq, 2018). I samband med detta lyfts kunskaper i programmering fram som något alla elever i grundskolan i Sverige ska få med sig. Från och med hösten 2018 är programmering ett obligatoriskt moment i undervisningen i grundskolan, redan från årskurs 1 (Skolverket, 2016; 2019a). Att ha en funktionell digital kompetens innebär att man både har tillgång till och förstår hur man kan använda den digitala utrustningen på ett adekvat sätt (Godhe, 2019). Genom att lära sig programmera kan eleverna få möjlighet att träna flera förmågor, så som kritiskt tänkande och matematisk problemlösning (Popat & Starkey, 2019). På det sättet erhåller de en digital kompetens som inte är ett mål i sig utan ett verktyg för lärande.

¹ Svenska Science Centers är en branschorganisation som arbetar för svenska vetenskapscenters gemensamma intresse; att väcka intresse för naturvetenskap, teknik, matematik hos barn, ungdomar och allmänhet.

Skolverket sammanfattar fyra aspekter av digital kompetens som tillsammans utgör grunderna för kunskaper alla elever ska få möjlighet att lära sig; *Att förstå digitaliseringens påverkan på samhället*, *Att kunna använda och förstå digitala verktyg och medier*, *Att ha ett kritiskt och ansvarsfullt förhållningssätt* samt *Att kunna lösa problem och omsätta idéer i handling* (Skolverket, 2017b). Det är den sista av dessa fyra aspekter som sätter fokus på programmering, då problemlösning med digital teknik bland annat hänvisas till programmering i ämnena teknik och matematik (Godhe, 2019). Programmering kan då bli en del av elevernas digitala kompetens.

1.1.1 Elevperspektiv

Denna studie har ett elevperspektiv och det är elevernas läroprocess som är av intresse. Jag har genom observation och intervjuer undersökt hur elever gör när de programmerar och analyserat deras läroprocess. Samtidigt kan jag aldrig ta en elevs plats. Det är därför viktigt att poängtera att det är min tolkning av elevernas perspektiv som redovisas. Eleverna betraktas alla som nybörjare då det är den typen av lektion de deltar i. För de flesta av eleverna i studien är programmering något de inte har arbetat med på lektionstid mer än vid något enstaka tillfälle. Hur mycket erfarenhet eleverna faktiskt har av programmering kan dock skilja sig åt då några elever provat programmering i andra sammanhang och vissa elever kan till och med ha programmerat mycket.

Studien är genomförd i brytningspunkten strax innan programmering blev en obligatorisk del av undervisningen i svensk grundskola. Det kan vara viktigt att man undersöker hur läroprocessen kan ta sig i uttryck redan hos de som är tidigt ute och tar till sig ny teknik. På detta sätt finns en möjlighet att sprida pionjärernas kunskap vidare och bana väg för de som är lite mer tveksamma (Jahnke & Kumar, 2014). Även om många lärare har genomfört programmeringsprojekt under samma tidperiod är det i denna studie elever som besöker SC som får ta rollen som pionjärer. Ett argument för detta är att där finns olika elever, från olika skolor och sammanhang. Elevperspektivet kan då breddas till att innefatta elever i olika åldrar och från olika skolor. Mot detta kan man ifrågasätta hur tillgängligheten för undervisning på SC ser ut i landet i stort, då det endast finns 19 SC.

Samtidigt har man i andra sammanhang och i andra länder undervisat i programmering i flera år. Det är därför motiverat att ta avstamp i tidigare studier som fokuserar på programmeringsundervisning för barn och unga

(exempelvis: Asad et al., 2016; Falcão et al., 2017). Detta avstamp ger bland annat insikt om olika typer av programmering och elevers attityd till programmeringsundervisning, något som är viktiga aspekter för studien.

1.1.2 Science center som lärmiljö

Lärmiljön där eleverna befinner sig i denna studie är således lektioner på SC. Det kan anses vara en adekvat miljö att använda i ett projekt om lärande i programmering då SC har lång erfarenhet av programmering för barn och unga (Svenska Science Centers, 2018). Detta nyttjades vid tidpunkten för studien, det vill säga redan innan programmering blev obligatoriskt i grundskolan, av skolor som bokade lektioner för elever och fortbildning för sin personal. SC är traditionellt sett en miljö som står mellan att vara en rolig fritidssysselsättning och en kunskapsförmedlare inom naturvetenskap och teknik (Kubacki et al., 2012). SC har haft, och har till stor del även fortfarande, rollen som informell lärandemiljö vilken kan jämföras med exempelvis museum (DeWitt & Osborne, 2007; Sasson, 2014). Då skolor utnyttjar SC till en del av undervisningen övergår SC från en informell lärandemiljö till vad som kan betecknas vara en semiformell miljö på gränsen till formell. De formella normerna för lärande i skolan möter de mer informella normerna för lärande på SC. Det kan därför uppstå intressekonflikter kring exempelvis kunskapsinnehållet under lektionerna.

1.1.3 Designorienterad teori

Genom att tillämpa ett designorienterat perspektiv (Kress & Selander, 2012; Selander, 2008; 2017) bidrar studien till insikter och förståelse om hur möjliga lärtillfällen designas och vad som händer med designen när elever ska lära sig att programmera. Det grundläggande intresset för genomförandet av denna studie är elevers möjligheter till lärande. Hur deras lärprocess ser ut och vad som händer när de ska (lära sig att) programmera. För att undersöka detta och för att förstå lärande i relation till de förutsättningar som skapas i olika miljöer och situationer används *Design för lärande* (Kress & Selander, 2012; Selander, 2008; 2017) både som teoretisk utgångspunkt och analysredskap. Utifrån detta perspektiv använder jag begreppen *design*, *iscensättning*, *transformationsprocesser* och *lärande* som utgångspunkt för att analysera och förstå min empiri. Teorin skiljer på *design för* respektive *design i* lärande, något som utvecklas längre fram i texten och som har betydelse för elevperspektivet i denna studie.

1.2 Syfte och frågeställning

Studiens övergripande syfte är att bidra med kunskap om hur elever designar sitt lärande i programmering. För att få en förståelse för hur elever designar både *för* och *i* sitt lärande studeras elevers olika sätt att lösa problem inom programmering. Då engagemang ofta lyfts fram som en viktig faktor för lärande, är det även intressant att få fördjupade kunskaper om hur engagemang kan relateras till hur elever designar i sin lärprocess. Därför undersöks i denna studie både elevers tillvägagångssätt och engagemang under programmeringslektioner på SC. Detta leder fram till studiens forskningsfrågor i de två artiklar som tillsammans bidrar till en gemensam bild av grundskoleelevers lärprocesser i programmering.

Följande frågor ställs om programmeringsundervisning för grundskolelever på SC:

- Hur ser lärprocessen för nybörjare i programmering ut?
- Hur kan förmågor, identifierade i olika tillvägagångssätt, förstås då de analyseras och jämförs med datalogiskt tänkande (CT)?

I den första artikeln analyseras hur de olika tillvägagångssätt som eleverna använder ger möjligheter att öva och använda förmågor som antas vara relevanta för programmering och digital kompetens.

Den andra artikeln svarar mot hela studiens syfte genom följande forskningsfråga:

- Hur blir olika typer av engagemang synliga under programmeringslektioner på SC?

I artikeln diskuteras även hur elevernas beteendemässiga, emotionella och kognitiva engagemang kan relateras till hur de designar i sin lärprocess.

I den sammanlagda studien diskuteras även den övergripande frågan hur elevers design för lärande i programmering kan beskrivas med bakgrund i observationer av och intervjuer med elever på SC.

1.3 Studiens disposition

Studiens inledande kapitel innehåller utöver en bakgrundsbeskrivning, syfte och forskningsfrågor, även en kort presentation av studiens teoretiska ramverk. I kapitel 2 fördjupas beskrivningen av bakgrunden till studien.

Två områden lyfts upp i denna del, digital kompetens i svensk grundskola-där lärande i programmering är en del, samt science centers roll för utbildning. Kapitel 3 ägnas åt internationell forskning om programmeringsutbildning och barn och ungas lärande i programmering. Kapitel 4 fortsätter med att beskriva den teoretiska bakgrunden, särskilt fokus läggs på modellen Learning Design Sequences. Vidare beskrivs även hur engagemang definieras och används för att analysera elevers lärandedesign. Kapitel 5 är metodkapitlet, där redogörs utöver urval och etiska aspekter, metoden för datainsamling och analys. I kapitel 6 ges en redogörelse för resultaten utifrån forskningsfrågorna i respektive artikel som utgör basen för denna studie. Dessa resultat diskuteras med utgångspunkt från syftet med hela studien i kapitel 7. Här redogörs även för vilka implikationer dessa resultat ger för att fortsatt utveckling av lärande i programmering samt förslag på fortsatt forskning.

2. BAKGRUND

För att ge en förståelse för den kontext detta arbete befinner sig i följer i avsnitt 2.1 först en bakgrund till den pågående digitaliseringen i samhälle och skola samt en bakgrund till hur och varför programmering implementeras i svensk grundskola. I detta avsnitt finns även en beskrivning av programmeringsundervisning för nybörjare samt en redogörelse av begreppet datalogiskt tänkande. I avsnitt 2.2 presenteras svenska science centers och dess roll i utbildningssammanhang.

2.1 Digitalisering och programmering i svensk skola

Kraven på att medborgare ska ha digital kompetens blir allt tydligare i dagens samhälle, både i internationell (Asad et al., 2016; Casey et al., 2018) och svensk kontext (Fleisher & Kvarnsell, 2017; Mannila, 2017; Regeringskansliet, 2015). Vilka begrepp som används och vad det innebär konkret kan dock skilja sig mellan olika länder och i olika sammanhang (Siddiq, 2018). I den svenska läroplanen används konsekvent begreppet digital kompetens (Godhe, 2019) och det är det begreppet som används i denna studie för att förklara varför grundskoleelever i olika åldrar ska lära sig att programmera.

2.1.1 Digital kompetens

Då studien görs i en kontext gällande svenska grundskoleelever utgår det som beskrivs som digital kompetens i denna studie från den sammanfattning som Skolverket gör i kommentarmaterialet, *Få syn på digitaliseringen på grundskolenivå* (Skolverket, 2017b). Det vill säga de fyra aspekter av digital kompetens som finns med i inledningen av denna studie: *Att förstå digitaliseringens påverkan på samhället, Att kunna använda och förstå digitala verktyg och medier, Att ha ett kritiskt och ansvarsfullt förhållningssätt samt Att kunna lösa problem och omsätta idéer i handling* (Skolverket, 2017b). Enligt dessa aspekter innebär digital kompetens att individen har en förståelse för hur digitaliseringen påverkar utvecklingen både på ett individuellt plan och i samhället i stort. Vidare krävs en förmåga att både använda digitala system och tjänster på ett för sammanhanget adekvat sätt och att förstå hur de fungerar. Digital kompetens innebär även att den enskilde har ett kritiskt förhållningssätt till medier och information och slutligen även har en förmåga att vara kreativ och lösa problem med stöd av digital teknik (Skolverket, 2017b). Dessa beskrivningar är relativt övergripande och den svenska läroplanen

för grundskolan saknar en beskrivning av hur progressionen av digital kompetens ska se ut (Siddiq, 2018). Det finns inte heller någon detaljstyrning om vad undervisningen ska innehålla eller hur det ska gå till. Då detta är överlämnat till varje skola eller till och med varje enskild lärare finns det en stor variation hur man tolkar begreppen och hur man undervisar om digital kompetens i praktiken (Siddiq, 2018).

2.1.2 Digitalisering av skolan

Samtidigt är det inget nytt med informations- och kommunikationsteknik (IKT) och digitalisering i svensk skola. Satsningar på IKT i grundskolan och gymnasiet startade redan under 70-talet med projektet *Datorn i skolan*. Det fanns redan då en politisk vilja att ge så många elever som möjligt tillfälle att pröva denna nya teknik (Rolandsson, 2015). Projekt har sedan avlöst varandra, på initiativ av Skolöverstyrelsen, Regeringen, Skolverket och andra. Vissa satsningar har varit riktade till personal, andra till elever och ytterligare andra till att öka andelen datorer i skolan. Från och med 2006 har så kallade en-till-en-satsningar, det vill säga en dator till varje elev, gjorts hos flera både privata och kommunala skolhuvudmän (Willermark, 2018). För en mer omfattande sammanställning kring införandet av digitalisering i skolan se exempelvis Fleisher och Kvarnsell (2017) eller Willermark (2018).

Fortfarande finns det dock en digital ojämlikhet, både när det gäller tillgång till digital utrustning och kompetens hos lärare och elever (Samuelsson, 2014). Det gäller i landet som helhet, men även inom samma huvudman och till och med inom samma skola. Det finns även implikationer på att undervisningsmetoderna inte har utvecklats i takt med att den digitala tekniken har introducerats i skolan (Rolandsson, 2015). För att utnyttja fördelarna med den tekniska utrustningen krävs en kombination av teknisk- pedagogisk- och ämneskunskap hos läraren (Pareto & Willermark, 2019). Något som inte alltid har gått hand i hand.

2.1.3 Programmering in på schemat

Under 2010-talet införde allt fler länder programmering i sina nationella läroplaner. Ett av de större projekten har gjorts i England där man sedan 2013 undervisar i ämnet Computing (Department for Education, 2013). I Sverige har man följt med i utvecklingen och Skolverket fick under våren 2016 uppdrag av regeringen att ta fram ett förslag på hur digital kompetens och programmering skulle förstärkas och tydliggöras i undervisningen (Skolverket, 2016). De första förslagen presenterades under konferenser

under våren 2016 där inbjudna intressenter från myndigheter, lärosäten och huvudmän deltog för att bidra med synpunkter (Andersson et al., 2017; Heintz et al., 2016). I dessa förslag förordades det att programmering skulle införas som en del av den obligatoriska undervisningen i befintliga undervisningsämnen, främst teknik och matematik och inte som i England, införas som ett eget ämne. Förslagen klubbades senare igenom.

I den nu gällande läroplanens första kapitel gällande skolans värdegrund och uppdrag skrivs det fram att skolan har ett ansvar för att alla elever ska få en ökad förståelse för digitaliseringen i samhället och för att alla elever ska ha möjlighet att utveckla digital kompetens (Skolverket, 2019a).

”Skolan ska bidra till att eleverna utvecklar förståelse för hur digitaliseringen påverkar individen och samhällets utveckling. Alla elever ska ges möjlighet att utveckla sin förmåga att använda digital teknik. De ska även ges möjlighet att utveckla ett kritiskt och ansvarsfullt förhållningssätt till digital teknik, för att kunna se möjligheter och förstå risker samt kunna värdera information. Utbildningen ska därigenom ge eleverna förutsättningar att utveckla digital kompetens och ett förhållningssätt som främjar entreprenörskap” (Skolverket, 2019a, s.7-8).

Som en del av den digitala kompetensen skrivs programmering fram som obligatoriska moment i det centrala undervisningsinnehållet. I kursplanerna för grundskolan i matematik och teknik presenteras olika aspekter av programmering. I matematikämnet lyfts likheten mellan programmering och algoritmer fram. Programmering kan då användas för att tydliggöra och berika matematikundervisningen. I tekniken är programmering mer inriktat mot hur man med den digitala tekniken kan styra föremål. Detta ger en antydning om att programmering är något som används i många olika sammanhang.

Ur kursplan i matematik, centralt innehåll för årskurs 4–6 (Skolverket 2019a, s. 57).

Hur algoritmer kan skapas och användas vid programmering. Programmering i visuella programmeringsmiljöer.

Ur kursplan i teknik, centralt innehåll för årskurs 4–6 (Skolverket 2019a, s. 294).

Att styra egna konstruktioner eller andra föremål med programmering.

Exemplen är hämtade från centralt innehåll för åk 4-6. För årskurs 1-3 respektive 7-9 är formuleringarna liknande, anpassade efter ålder. Det

finns även en viss progression i hur formuleringarna preciserar programmering mellan de olika stadierna. I årskurs 4-6 ska eleverna exempelvis i matematik programmera i visuella programmeringsmiljöer, för årskurs 7-9 anges i *olika* programmeringsmiljöer, vilket innebär att även textbaserat programspråk kan vara lämpligt att använda.

Programmering är således ett ämnesövergripande område, vilket gör att det alltid finns en risk att ingen tar fullt ansvar för att driva undervisningen framåt, något Kjällander et al. (2016) påpekar i en forskningsöversikt gällande programmering. Detta kan motverkas då det i kursplanerna till läroplanen (Skolverket, 2019a) alltså finns en differentiering vad som skrivs fram i de olika ämnena. Vid en jämförelse mellan de olika ämnena fokuseras det inom matematiken på hur algoritmer kan skapas och användas vid programmering. Det är centralt då algoritmer kan beskriva matematiska handlingar på ett sätt som kan översättas till en kod som datorn förstår. Inom tekniken kopplas programmering istället mer till hur olika föremål kan styras och regleras med hjälp av programmering. Det vill säga ett större fokus på vad man kan göra med digital teknik, exempelvis hur sensorer kan användas för att styra och kontrollera tekniska lösningar. Denna differentiering gör att både lärare i matematik och i teknik har ett ansvar för elevernas lärande i programmering. Det finns ibland ett förgivettagande gällande transfereffekter, det vill säga att det man lär sig i det ena ämnet kan överföras till det andra (Åkerfeldt et al., 2018). Något som dock har varit svårt att visa i vetenskapliga studier.

Grundskolans kunskapskrav i matematik och teknik har inte reviderats till att varken gälla digital kompetens eller specifikt programmering utan har kvar samma formuleringar som tidigare (Skolverket, 2019a). Det innebär att digital kompetens inte ska bedömas separat utifrån specifika kunskapskrav utan ses som en kompetens som är nödvändig för att förvärva annan kunskap (Godhe, 2019). Det är rektors och lärares ansvar att alla elever får möjlighet att utveckla sin digitala kompetens genom bland annat programmeringsundervisning och därför är det viktigt att de som undervisar har kompetens att fullgöra detta ansvar (Åkerfeldt et al., 2018).

Vid tidpunkten för datainsamlingen till denna studie var revideringen av läroplanen ännu inte fastställd. Därför inriktas heller inte studien explicit på kunskapsinnehåll i matematik eller teknik. Programmering betraktas i

studien alltså istället vara en del av den digitala kompetens alla elever ska ha tillgång till.

2.1.4 Undervisning för nybörjare i skolmiljö

Trots att man i många andra länder har arbetat med programmering i skolan är det ändå relativt nytt med programmering för elever i grundskoleålder i Sverige. Det innebär att man provar sig fram på olika sätt (Mannila, 2017). Det kan tänkas naturligt att använda någon form av digital utrustning men det går även att undervisa programmering utan digitala verktyg (jfr Åkerfeldt et al., 2018). Spelkort, papper och penna eller varför inte pinnar och kottar i skogen är material som används för att introducera begrepp som villkor, loopar och variabler. För den allra första undervisningen i programmering, kan det vara bra att förstå att grunderna i alla typer av program är att det bygger på stegvisa instruktioner. Ett av de mer klassiska exemplen för att förklara hur programmering går till kan exempelvis vara att man be någon att skriva en "te-algoritm" (Mannila, 2017). Jag återberättar exemplet nedan för att åskådliggöra kravet på noggrannhet i varje steg:

För att förstå hur noggranna instruktioner ett datorprogram behöver, räcker det inte i te-algoritmen att skriva "koka te". Varje steg behöver definieras för att man verkligen ska kunna få sin välbehövliga kopp te. Från att ta fram en kastrull, via att sätta på kranen (och stänga av den igen), ställa grytan på spisen, sätta på plattan och vänta tills det kokar, till att lägga i tepåsen i koppen och fylla på det heta vattnet. På samma sätt fungerar ett datorprogram, varje steg måste definieras för att en önskad funktion ska bli verklighet (fritt återberättat från Mannila, 2017).

Även om det alltså går att göra mycket utan dator för att förstå grundläggande principer för programmering kan det behövas en djupare reflektion kring designen av undervisningen när man sen ska integrera de digitala verktygen. Det är en blir en dubbel utmaning då designen behöver innehålla både det digitala verktyget och ämnesinnehållet samtidigt (Jahnke & Kumar, 2014; Selander, 2007). Vad som är viktigast, att förstå hur utrustningen fungerar, eller att förstå vad man kan lösa för typ av problem med hjälp av programmering kan därför diskuteras. Pareto och Willermark (2019) visar hur alla delar är viktiga för att undervisning ska fungera, både de tekniska och de mer allmändidaktiska. För lärare som inte är vana att undervisa i programmering kan det vara en puckel att komma

över, det vill säga att ha en förmåga att designa för en undervisning som tar hänsyn till alla aspekterna.

Inom programmering finns flera olika typer av kod, eller programspråk, som har olika användningsområden. Följaktligen har formuleringen att *lära sig programmera* olika betydelser för olika personer och det kan därför vara nödvändigt att definiera vad det är för avsikt med programmeringen. Det kan finnas en skiljelinje gällande för vad man avser är programmering beroende på vilken typ av kod man använder. Inom undervisning i programmering för nybörjare används både textbaserad kod och visuell kod. Den textbaserade programmering benämns ibland som att man programmerar på riktigt (Skansholm, 2018). Det vill säga att man genom att lära sig programmera med Python, eller något annat språk på mer hög eller så kallad *formell nivå*, lär sig arbeta mer som en programmerare (Downey, 2015). På formell nivå är programmeringsspråket likt vanligt talat eller skrivet språk, men varje funktion har sina uttryck. Det kan därför vara många uttryck som behöver läras in för att kunna använda dem på ett funktionellt sätt. I visuell kod/blockkod är dessa funktioner redan förprogrammerade och behöver inte definieras varje gång vilket kan underlätta för att en nybörjare snabbt ska kunna få ett program att fungera.

2.1.5 Förutsättningar för att undervisa i programmering

Redan innan kursplanerna fick sitt slutgiltiga innehåll presenterat, påbörjades i hela landet olika projekt för hur programmering skulle implementeras i undervisningen. Arbetet intensifierades sedan kursplaneförändringarna gällande programmering fastslogs sommaren 2017. Enskilda lärare har tagit ett stort ansvar och projekt i programmering på grundskolan har genomförts exempelvis bland annat i Halmstad (Halmstad kommun, 2019) och Skellefteå (Brising, 2019). Skolor har även sökt kompetens hos andra aktörer såsom lokala Komtek, Kommunal entreprenör- och teknikskola (Komtek, u.å.) och Makerspace (Makers of Sweden, u.å.). På de platser i landet där det finns science center har dessa tagit en aktiv roll för implementering av framförallt programmeringsundervisningen i teknikämnet (Svenska Science center, 2018). En bidragande orsak till dessa samarbeten är att hos dessa organisationer finns en erfarenhet av programmering för barn och unga genom de kursverksamheter som erbjuds som fritidsaktiviteter. Det finns dessutom redan både lokaler och utrustning på plats. De olika aktörerna

har dessutom ofta stor möjlighet att anpassa verksamheten och snabbt ställa om sin undervisning till aktuella teman.

Samtidigt finns ett mer allmänt behov av kompetensutveckling hos lärarkåren (Fleischer & Kvarnsell, 2017). Efter att programmering skrevs in i kursplanerna har Skolverket gjort satsningar för att stärka lärarnas kompetens inom området. Skolverket använde bland annat SC:s som resurs vid konferenser kring implementeringsarbetet av programmering i teknikämnet i grundskolan (Skolverket, 2017a). Under 2019 lanserades webbkurser och Skolverkets lärportal har allt eftersom kompletterats med moduler och filmer inom programmeringsundervisning (sökord programmering på skolverket.se). Hela satsningen med programmering är enligt Skolverket en del av skolutvecklingsprogrammet *Digitalisering*.

2.1.6 Datalogiskt tänkande

Begreppet *datalogiskt tänkande* (på engelska; *computational thinking*, CT) används inte i den svenska läroplanen, varken i kursplanen för teknik eller för matematik (Skolverket, 2019a). Det är också tydligt att CT ännu inte har en exakt samstämmig definition (Zhang & Nouri, 2019). Ändå används begreppet ofta i debatten om programmeringsutbildning i Sverige (CETIS, 2017; Heintz et al., 2016). Popat och Starkey (2019) ställer frågan om man lär sig att koda eller kodar för att lära sig. Liknade fråga kan ställas angående CT, det vill säga huruvida man lär sig CT eller om man använder CT för att lära sig. Åkerfeldt et al. (2018) påpekar just detta att CT inte skrivs fram i styrdokumentet men lyfter ändå fram begreppet som ett sätt att förklara hur man inom datavetenskap och programmering definierar och delar upp problem. Även Mannila (2017) beskriver CT som ett paraplybegrepp över förmågor som används vid och även kan tränas med hjälp av programmering. Det finns alltså argument både för och emot att använda begreppet i samband med programmeringsundervisningen för barn och unga. Eftersom CT kan sammanfattas till att huvudsakligen innefatta olika sätt att tänka vid problemlösning, något som är viktigt för att kunna programmera (Cetin & Dubinsky, 2017; Durak & Saritepeci, 2018; Wing, 2006), ser jag att det kan finnas vinst i att relatera och jämföra de förmågor elever använder till detta begrepp.

Ibland delas CT upp i övergripande idéer respektive strategier för att lösa uppgiften, på engelska *concepts* and *approaches* (Kalelioğlu et al., 2017). Utgångsläget är då, att för att bli bra på något som programmering, måste

människor utveckla en strategi för att effektivt sätta sina tankar i handling. I dessa fall blir exempelvis att dela upp problem i mindre delar och lösa ett steg i taget (på engelska, decomposition) en övergripande idé. Tinkering (begreppet används även på svenska) och felsökning (på engelska, debugging) räknas istället som strategier. Tinkering kan betraktas vara ett lekfullt sätt att lära sig på, utan att från början ha ett direkt mål för sin aktivitet (Kalelioğlu et al., 2017). Felsökning är något vi gör dagligen, att hitta misstag och sedan rätta till. När felsökning är en del av CT söker man efter buggar systematiskt och strategiskt (Kalelioğlu et al., 2017). Wing (2006) påpekar att tänka som en programmerare, det vill säga att använda CT, är mer än att kunna programmera, det är komplext och kräver förmåga att tänka abstrakt i flera nivåer.

I denna studie ses CT som övergripande kompetens som används vid programmering. Att programmera är inte det samma som CT, men det kan underlätta att programmera om man använder CT. På detta sätt vävs de övergripande idéerna och strategierna samman till ett antal förmågor. Först och främst är CT något som är beroende av människor, det handlar om hur människor tänker och hur de analyserar problem (Wing, 2006). Det handlar om att organisera och strukturera på ett logiskt sätt för att komma fram till bra lösningar på olika typer av uppgifter (Tran, 2019). I Zhang och Nouris sammanställning (2019) av olika definitioner av CT anges begreppet decomposition som en viktig aspekt. Det innebär alltså att dela upp problem i mindre delar för att sedan lösa hela problemet en bit i taget. En annan särskilt viktig del av CT, är förmågan att förstå och använda algoritmer (Durak & Saritepeci, 2018; Tran, 2019; Zhang och Nouri, 2019). Det vill säga att tydligt kunna definiera varje steg och sedan automatisera lösningar därefter. CT innefattar även inte minst individens kreativitet när det gäller att hitta nya vägar till lösningar (Durak & Saritepeci, 2018; Kalelioğlu et al., 2017). Det ger tillfälle till engagemang för att designa och skapa. CT handlar även om uthållighet (Kalelioğlu et al., 2017). En förmåga som är nödvändig då det kan fordra mycket tid att kunna skapa något nytt och se nya möjligheter och att man ofta får börja om många gånger.

2.2 Science centers

Ett Science Center är en interaktiv anläggning där fokus ligger i att väcka intresse och nyfikenhet hos barn och ungdomar för matematik, naturvetenskap och teknik (Svenska Science Centers, 2018). SC:s är inget

svenskt fenomen utan finns på många platser i världen, mitt i det moderna samhället (se exempelvis Pedretti, 2004; Sasson, 2014; Streicher, 2016). Syftet med SC är att lyfta allmänhetens engagemang för och kunskaper i, relevanta ämnen med anknytning till naturvetenskap och teknik (Achiam & Sølberg, 2016). Anläggningar finns i de flesta världsdelar och knyter samman sina verksamheter i olika internationella nätverk.

2.2.1 Föreningen Svenska Science Centers

2017, när datainsamlingen till föreliggande studie är gjord, fanns det 19 SC:s som tillhörde föreningen Svenska Science Centers. Anläggningarna varierar i storlek, är spridda över hela landet, från Malmö i söder till Luleå i norr och har olika typer av huvudmän och finansieringslösningar. Exempel på huvudmän är universitet, kommunägda bolag, del av kommunal förvaltning eller stiftelser. För att vara medlem i Svenska Science centers krävs att samtliga uppställda kriterier i föreningens stadgar är uppfyllda (Svenska Science Centers, 2017). I dessa stadgar finns bland annat angivet att en väsentlig del av verksamheten på SC ska vända sig till allmänheten samtidigt som ämnesövergripande program för skola, förskola ska erbjudas. För möjligheten att söka statligt verksamhetsbidrag granskas svenska science centers årligen av Skolverket med avseende på verksamhet och kvalitet (Skolverket, 2019b).

Svenska Science centers tar årligen emot ca 2 miljoner besökare, varav ca 400 000 är elever från de flesta av Sveriges kommuner. Det vill säga att var femte besökare på SC är en skolelev som får en del av sin undervisning på SC. Vidare organiserar Svenska Science Centers årligen kompetensutveckling för ca 22 000 lärare och pedagoger i programmering, teknik, naturvetenskap och matematik (Svenska Science Centers, 2018).

2.2.2 Lärmiljö i ett spänningsfält

Ett SC är varken en skola eller ett museum, det befinner sig snarare i spänningsfältet mellan den formella utbildningen i skolan och en rolig fritidsaktivitet (Kubacki et al., 2012). Om vi i det sammanhanget benämner skola som en formell lärandemiljö och lärande som sker i det dagliga livet som informell lärandemiljö (Sasson, 2014) blir SC något däremellan, det vill säga en semiformell lärmiljö. Denna distinktion kan även göras genom att se på hur miljön är designad för lärande och huruvida det är obligatoriskt att delta i de aktiviteter som erbjuds. SC är designade för lärande, främst inom naturvetenskap och teknik. Anläggningarna är tillgängliga för frivilliga besök på egen hand. Samtidigt är de också delvis

anpassade för mer obligatoriska undervisningstillfällen exempelvis när anläggningarna bokas av skolor. Sådana bokade besök kan gälla både för grundutbildning och högre utbildningar. Det leder alltså till en semiformell miljö som ibland är frivillig och ibland obligatorisk beroende på hur besöket är organiserat. Det som kännetecknar didaktiken på SC är att besökarna utmanas genom praktiska aktiviteter att undersöka för besökaren nya områden. Det kan i sin tur ge vidare möjlighet till reflektion för fortsatt lärande (Streicher, 2016). På så sätt är tanken att tröskeln till att lära ska kännas låg, även om man inte har någon tidigare erfarenhet av det aktuella området.

Även om experimentstationerna och utställningarna på SC är pedagogiskt utformade är det inte alltid som det finns en pedagogisk handledningen i ”levande form”, det vill säga personal som kan guida besökarna runt. Därför kan det finnas ett behov att stödja de lärare som väljer att besöka ett SC tillsammans med sina elever, i hur de bäst kan utnyttja de miljöer som finns där. DeWitt och Osborne (2007) sammanfattar tidigare forskning om skolbesök på SC med att det finns vinster i att lärarna i förväg bekantar sig med anläggningen. Detta för att kunna planera både besöket så att de planerade aktiviteterna stödjer gällande läroplansmål, men även för att kunna planera både för- och efterbete som kan göras på skolan. DeWitt och Osborne (2007) menar att trots att dessa rekommendationer för god praxis inför SC-besök har funnits länge, finns det fortsatt utvecklingspotential inom området. Martin et al. (2016) lyfter fram ett antal aspekter gällande besökares engagemang och möjlighet till lärande som man bör ta hänsyn till vid planering av olika undervisningsprogram på SC och liknande. Det handlar om hur man ger möjlighet till att ta del av instruktioner på olika sätt, att instruktörer stöttar besökarnas interaktion med olika typer av resurser samt att medvetet planera för att undervisningsprogrammen ska väcka känslor.

På flera SC anordnas styrd verksamhet för förbokade grupper såsom planerade lektioner som en del av skoldagen (Martin et al., 2016; Sasson, 2014; Svenska Science Centers, 2018). SC erbjuder då lektioner och aktiviteter framförallt i ämnen som ibland anses svåra att förstå och därför kan behöva erfaras på andra sätt än vad som är möjligt i ett vanligt klassrum. Det kan även gälla områden där det krävs speciell utrustning som kan vara dyr att tillhandahålla på en enskild skola. Vid bokande lektioner får instruktörerna på SC en mer aktiv roll att styra lärandeinhåll i en viss riktning än om eleverna undersöker på egen

hand i utställningen med de olika experimentstationerna. Hur besök på SC utnyttjas av skolor med yngre elever som en del av undervisningen har beskrivits av bland Shaby et al. (2017). De analyserar hur elever upplever besök på SC och det är tre områden de lyfter fram som betydelsefulla för ett bra och lärorikt besök; att det finns tid för aktiviteter där eleverna själva får styra vad de vill undersöka, att det är lätt att förstå hur man ska använda utrustningen och de olika experimentstationerna, samt att eleverna tätt förknippar det som de har lärt sig med de instruktioner de fått från SC-personalen (Shaby et al., 2017).

En del av den forskning som finns om SC och andra liknande miljöer, så som museer, har traditionellt handlat om besökarna och vad de gör eller vad de lär sig när de undersöker olika fenomen i utställningarna (Davidsson, 2008). Till exempel har undersökningar visat hur besökare på SC:s uppfattar att det är en stimulerade miljö och att de har en möjlighet att lära sig mer om naturvetenskap och teknik (Falk & Needham, 2011; Pedretti, 2004). Andra studier analyserar huruvida naturvetenskapliga utställningar kan hjälpa till att fylla det gap som kan finnas mellan de lärandes förståelse och olika naturvetenskapliga sammanhang (Piqueras et al., 2008). Streicher (2016) lyfter fram möjligheten att även universitetsstudenter har stor nytta av att ta del av de lärmiljöer SC erbjuder. Genom att praktiskt prova, utan att det blir trivialt, kan de interaktivt erfara olika typer av fenomen och på det sättet förbereda sig inför mer teoretiska moment. Vidare inriktas en del forskning på hur elevers motivation att lära mer om naturvetenskap och teknik påverkas av besök på SC (Itzek-Greulich & Vollmer, 2017; Kubacki et al., 2012). Här kan det finnas fördelar genom att SC kan fungera som en bra igångsättare när eleverna har möjlighet att pröva sina kunskaper praktiskt. Eleverna kan också känna att de har större frihet då de inte blir betygssatta av SC-instruktörerna.

Det finns implikationer på att traditionell undervisning i skolan har större möjlighet att fördjupa kunskaper inom de mer teoretiska avsnitten som tillhör de aktuella kunskapsområdena (Itzek-Greulich & Vollmer, 2017). Sasson (2014) menar att det därför finns en anledning att fortsätta utveckla pedagogiken så att lärsituationen inte stannar vid att fokusera på intressen och attityder utan även koncentrerar mer på vilken kunskap eleverna får med sig. Sasson (2014) påpekar att det finns en risk att SC fokuserar för mycket på artefakter istället för pedagogiska principer. Därför kan

föreliggande studie ge ett bidrag till att lyfta dessa frågor då syftet just är att studera elevers läroprocess när de programmerar på SC.

2.2.3 Programmeringsverktyg på SC

Som tidigare nämnts är en av huvudanledningarna för skolor att boka in lektioner på SC, att det på skolan inte finns utrustning lämplig för det aktuella kunskapsområdet. När det gäller programmering kan man på SC använda samma utrustning till många olika grupper och personalen har möjlighet att fördjupa sig i hur olika programmeringsverktyg kan användas på olika sätt. Nedan följer en kort beskrivning av de programmeringsverktyg som användes på SC vid tiden för studien.

Först och främst behövs så kallad hårdvara, det vill säga utrustning som man kan "ta på". I första hand gäller det då tillgång till dator eller surfplatta. I förekommande fall räknas även undervisningsrobotar och annan utrustning till hårdvaran. Till det behövs någon form av mjukvara, det vill säga programvara till de program som används under lektionerna. De programvaror som användes är företrädesvis baserade på så kallad visuell blockprogrammering. Det vill säga att man exempelvis genom att placera förprogrammerade block i rätt ordning på datorn eller surfplattan, kan styra en fysisk robot eller konstruera ett spel visuellt på skärmen. Blocken kan liknas vid pusselbitar som justeras efter vad användaren vill utföra. Användaren kan se hela programmet på skärmen och det är möjligt att justera varje block för sig. Under de observerade lektionerna användes huvudsakligen programvaran Scratch samt LEGO®s egna programvara till LEGO®-robotar.

Scratch är en webbaserad programmeringsplattform utvecklad för barn. Scratch kan användas som introduktion till programmering men även för att lösa mer avancerade uppgifter. Under de observerade lektionerna användes Scratch till att programmera ett enkelt spel på dator. Scratch JR, är utvecklad för yngre barn och används istället på en surfplatta. Scratch och Scratch JR kan användas online eller laddas ner till datorn respektive surfplattan gratis.

De programmerbara robotarna som användes under den observerade lektionerna var LEGO® EW3 och LEGO® NXT. De är olika versioner LEGO®s robotar anpassade för lek och undervisning. Varje programmeringsblock i mjukvaran styr en viss funktion och eleverna kan då programmera för att till exempel styra robotarna att köra längs en

utstakad bana eller med hjälp av olika sensorer utföra olika typer av uppdrag.

Andra verktyg som användes på SC är Makey Makey och Arduino. För Makey Makey används vanligtvis en färdig programvara, med Arduino skriver eleverna koden själva med den speciella Arduino-programvaran. Slutligen används Kojo, och det är det enda verktyget, av de som användes under de observerade lektionerna, som styrs helt med skriven kod istället för blockprogrammering. Kojo är baserat på programmeringsspråket Scala. Kojo kan laddas ner gratis och finns på svenska.

För mer information om de olika programmeringsverktygen se respektive hemsida vilka finns listade i appendix 1.

3. TIDIGARE FORSKNING

För att få en bakgrund till området lärande och engagemang i programmeringsundervisning sammanfattas i de följande avsnitten tidigare forskning om de mest tillämpliga områdena för arbetet med studien, det vill säga programmeringsundervisning för barn och unga. Inom detta fält redovisas för aktuell forskning om lärande i programmering, begreppet datalogiskt tänkande och dess betydelse för programmering samt vikten av engagemang vid lärande.

Med hjälp av ett antal sökord anpassade till studiens syfte har jag systematiskt sökt litteratur i universitetsbibliotekets databaser, främst i ERIC (Educational Resources Information Center) som är en databas med referenser till pedagogik och psykologi. Allt eftersom skrivarbetet har fortskridit har återkommande sökningar gjorts. Gällande relevant forskning inom programmeringsundervisning har jag företrädesvis använt sökorden *programming*, *programming education*, *computational thinking*. Dessa ord har jag kombinerat med sökorden *pupils*, *students*, *learning*, *teaching*, *engagement*, *elementary school*, *primary school*, *compulsory school*, *science centre* och *science museum*.

3.1 Hård och mjuk programmering

Det finns, eller i alla fall har funnits, en vedertagen kultur om vad som är rätt eller fel sätt att arbeta och lösa problem med programmering. Den traditionella uppfattningen om programmering är att det handlar om tekniska och matematiska aktiviteter och att det krävs logiskt tänkande för att programmera. Pionjärer till att utmana detta sätt att tänka inom programmering var Turkle och Papert (1990). De menade att detta instrumentella sätt att se på programmering kunde riskera att utesluta många elever från att lära sig att programmera. Genom ett antal exempel visade de hur elever, både motsvarande yngre och äldre grundskoleelever, använde vad de säger "mjuka" sätt att programmera, istället för det traditionella "hårda". Turkle och Papert (1990) såg en skillnad på vilka sätt flest pojkar respektive flickor använde. De menade dock redan då, det vill säga för 30 år sedan, att det viktigast var inte är skillnaden mellan könen utan att lyfta fram olika sätt att arbeta.

I Turkle och Paperts (1990) artikel finns flera intressanta exempel, bland annat en 9-årig elev som fått i uppgift att programmera en legorobot att röra sig en viss sträcka. Det mest logiska sättet skulle ha varit att använda och programmera robotens hjul och motor på ett konventionellt sätt. Eleven valde i stället att ge roboten fötter genom att montera hjulen liggande och istället för att låta motorn driva hjulen lät eleven motorns vibrationer roboten att gunga eller ”gå” framåt. Programmeringen blev inte mindre avancerad, bara annorlunda.

Det som Turkle och Papert (1990) visade, att man genom att tillåta olika sätt att tänka och se att problem kan lösas på olika sätt, var något som de menade kunde göra programmering mer tillgänglig för fler. De typiska hårda respektive mjuka sätten att programmera ansågs ha utmärkande egenskaper, både vad gäller hur arbetet organiserades och förhållningssättet hur man såg på utrustningen. Hårt, abstrakt, systematiskt och distanserat eller mjukt, konkret och med en känsla av närhet. Båda dessa förhållningsätt har visat sig ha ett värde och det mer mjuka sättet har på senare år även blivit mer vedertaget inom programmering genom de visuella programspråken.

3.2 Visuell kod eller textbaserad kod

Den övervägande forskningen under 2010-talet kring programmering för elever i grundskoleålder rör programmering med visuell- och/eller blockbaserad kod (till exempel; Lewis, 2011; Sáez-López et al., 2016). Anledningen till det är sannolikt att det är mer tilltalande för unga elever att arbeta med färg och form och att det är mer lättillgängligt än att skriva textbaserad kod, något som bland annat bekräftas av Kalelioğlu (2015). Mycket forskning inom programmering för barn och unga kretsar kring hur programmet Scratch kan användas för att förstå grunderna i programmering (exempelvis Portelance et al., 2016; Ouahbi et al., 2015). Det är helt enkelt ett av de mest använda programmeringsverktygen för barn. Scratch är, som beskrivet tidigare, en webbaserad programmeringsplattform med verktyg utvecklade för barn. Då många använder plattformen är det också angeläget att forska om.

Ett exempel på denna typ av forskning är en studie som undersöker vilken typ ”pusselbits-block” elever helst föredrar att använda när de skapar egna projekt på surfplatta genom att programmera i Scratch JR (Portelance et al., 2016). Över 900 projekt gjorda av barn/elever i åldrarna förskola till

och med årskurs två analyserades. De block som användes mest av alla elever var de block som gör att avataren (figuren som syns på skärmen) rör på sig. När de såg avataren agera på skärmen fick eleverna en bekräftelse på att programmet motsvarande det de programmerat (Portelance et al., 2016). Resultaten i studien visar också på en signifikant skillnad gällande hur de yngre eleverna använde fler stopp-block än de något äldre eleverna som i sin tur använde fler kontrollblock. En tolkning Portelance et al. (2016) gör av detta resultat är att de yngre barnen tilltalades av symmetrin stopp-blocket gav, det vill säga att det fanns en start på programmet, en händelse och ett slut. De något äldre eleverna i årkurs två såg exempelvis istället möjligheten med att låta ett program fortsätta utan något slut, genom att låta ett kontrollblock repetera programmet flera gånger, det vill säga göra en loop. Sammanfattningsvis drar Portelance et al. (2016) slutsatsen att eleverna använde fler avancerade block ju äldre de var. De menar därför att Scratch JR kan vara ett lämpligt att använda av barn i lågstadieåldern för att lära sig grunderna för programmering.

Det kan ifrågasättas huruvida forskning om programmering för barn och unga handlar om lärande eller om det mer är attityder och uppfattningar om programmering. När det gäller Scratch har exempelvis Kalelioğlu och Gülbahar (2014) undersökt hur elevers problemlösningsförmåga utvecklades när de under fem veckor arbetade med programmet. Även om studien visar på en liten förbättring av elevernas *självförtroende* gällande problemlösning, visar samtidigt resultaten inte på någon signifikant förbättring av elevernas förmåga att lösa problem. Ur ett elevperspektiv menar dock Kalelioğlu och Gülbahar (2014) att resultaten är positiva i den meningen att eleverna gillade programmering och att de ville lära sig mer. I en annan studie av Kalelioğlu (2015) undersöktes hur programmeringsplattformen code.org kan användas för att undervisa i programmering. Den kvantitativa delen av studien visar inte på några signifikanta förbättringar av elevernas reflektiva problemlösningsförmåga. Den mer kvalitativa delen av undersökningen då eleverna intervjuades i fokusgrupper visar dock på att programmeringsundervisningen verkade uppskattas av elever i grundskoleåldern och att de ansåg att plattformen var lätt att använda (Kalelioğlu, 2015). Kalelioğlu (2015) drar därför slutsatsen att elever ändå bör introduceras i programmering tidigt då eleverna under veckorna de undervisades i programmering fick en positiv attityd. I förlängningen menar Kalelioğlu (2015) att det kan leda till att eleverna inte bara blir konsumenter av digital teknik utan även kan bidra som producenter.

Visuell blockprogrammering kan ibland utöver att användas inom matematik och teknik även användas för att förstärka och ge möjlighet till att uttrycka sina kunskaper inom andra ämnen. I en kvasiexperimentell studie av Sáez-López et al. (2016) som pågick under två år användes Scratch av elever i årskurs 5-6. De 107 eleverna i studien programmerade med Scratch i samhällsorienterade ämnen och bild. Deras resultat jämfördes med en kontrollgrupp som undervisades mer konventionellt. Resultaten från studien visar att eleverna uppfattade undervisningen med Scratch som positiv och motiverande. Samtidigt som de erhöll kunskaper i programmering ökade de även sin förmåga inom både historia och bild (Sáez-López et al., 2016).

Forskning om mer textbaserad programmering för barn och ungdomar fokuserar ofta på en jämförelse av textbaserad programmering och blockbaserad visuell programmering. Mladenovic et al. (2018) har undersökt hur olika typer av programspråk påverkar elevers möjlighet att lära sig programmering. Deras studie genomfördes med 207 elever på grundskolenivå och man använde då tre olika programspråk, Scratch, Logo och Python. I studien valde man ut att jämföra hur eleverna klarade av att programmera med loopar, det vill säga att vissa instruktioner ska upprepas flera gånger. För att kunna jämföra elevernas resultat konstruerades motsvarande tester för de olika programmeringsspråken. Resultatet visar att eleverna klarande uppgifterna bättre när de använde ett blockbaserat programmeringsspråk jämfört med textbaserade programmeringsspråk. Mladenovic et al. (2018) förklarar sina resultat med att då programmering i sig är en abstrakt aktivitet kan det vara svårt för eleverna att se logiken i det de gör när de även behöver koncentrera sig på att komma ihåg rätt kommandon i en textbaserad miljö. Då kan svårigheterna med uppbyggnaden av exempelvis en loop ligga i att använda rätt kommandon, inte i att förstå hur loopen ska användas (Mladenovic et al., 2018).

Alla programspråk har dock både styrkor och svagheter. För att få svar på hur olika typer av kod i undervisningen kan påverka lärande i programmering använde Lykke et al. (2015) tre olika designar för undervisning. Traditionell lärarledd programmeringsundervisning med textbaserat kodspråk, problembaserad undervisning samt problembaserad undervisning med hjälp av visuell programmering med lego-robotar. Resultatet visar bland annat på att undervisningen med robotarna var engagerande och motiverande. Robotundervisningen upplevdes dock också frustrerande på grund av bland annat praktiska problem när robotarna inte fungerade som det var avsett. Framförallt kände sig

studenterna osäkra på om de lärde sig rätt saker när de programmerade på andra sätt än det traditionella. Undervisningen med textbaserade kod gjorde däremot att studenterna kände sig mer säkra gällande teoretisk förståelse. Samtidigt upplevde sig studenterna mer stressande. Både på grund av sin begränsade erfarenhet av programmering och att det inte fanns möjlighet att samarbeta på samma vis när allt genomfördes teoretiskt. Studien visade sammantaget att ingen av inlärningsdesignerna var helt tillfredsställande, det fanns för- och nackdelar med alla tre (Lykke et al., 2015).

När bara visuell programmering undersöks är tonen ofta positiv, vid jämförande studier där visuell kod ställs mot textbaserad kod kan resultaten vara mer nyanserade. För att återknyta till kapitel 2.1. om digitalisering och programmering i svensk skola, kan det finnas anledning att fundera över vad syftet med undervisningen är och vilket ämne som ska undervisas för att helt enkelt bestämma vilken typ av programmering som fungerar bäst i sammanhanget. Både visuell och textbaserad programmering har sina för- och nackdelar och fungerar beroende på sammanhanget olika bra.

3.3 Datalogiskt tänkande och programmering

I en forskningsöversikt har Buitrago Flórez et al. (2017) sammanställt resultaten av 92 studier om programmeringsundervisning i olika åldrar. Studierna som ingår i sammanställningen behandlar olika typer av programmeringsverktyg och man ser både utmaningar och fördelar med olika sätt att undervisa. Slutsatsen Buitrago Flórez et al. (2017) drar är att man inte bör undervisa programmering enbart för att eleverna ska lära sig att skriva kod. Undervisningen i programmering bör i stället inrikta sig på olika typer av förmågor som kan relateras till datalogiskt tänkande (CT). De menar vidare med utgångspunkt i resultaten av de sammanställda studierna att elever bör lära sig programmering och CT redan i grundskolan. Samtidigt framhåller de att det fortsatt viktigt att fokusera på CT även i universitetskurser då det är något som gynnar alla.

Programmeringsundervisning för nybörjare kopplas alltså ofta samman med CT. Zhang och Nouri (2019) har därför gjort en systematisk översikt över hur undervisning i just Scratch kan gynna CT. 55 studier valdes ut där åldern på deltagarna motsvarande förskola- grundskola årskurs nio, studierna var empiriska, samt där att studierna bedömde hur CT-förmågor

gynnas av Scratch. Översikten visar att det fortfarande finns en diskrepans med vad olika forskare avser med CT, samtidigt menar författarna att detta kan leda till en ökad förståelse av de olika förmågor som kan läsas in i CT. Dessutom kan just att CT har en vid betydelse ge en insikt i vilka svårigheter och utmaningar elever kan möta när de programmerar (Zhang & Nouri, 2019).

Parallellt som det alltså finns skilda åsikter om vad CT är, diskuteras det om CT är knutet till något visst ämne eller om CT enbart är användbara förmågor inom problemlösning och i samband med programmering. Durak och Saritepeci (2018) visar att exempelvis elevers både intresse för och kunskaper inom matematik i hög grad korrelerar med deras CT förmågor. I undersökningen, som innefattar 156 elever i årskurs fem till och med motsvarande gymnasiet, jämfördes elevernas CT-förmågor i förhållande till olika områden. Bland dessa områden fanns elevers sätt att tänka, elevers inställning till matematik och naturvetenskap samt elevernas kunskaper inom matematik och naturvetenskap. Resultaten visar bland annat att hög CT-förmåga kan kopplas samman med olika sätt att tänka. Vidare visar resultaten att elever med goda resultat och positiv attityd till matematik visade även på goda CT förmågor (Durak & Saritepeci, 2018). Kong et al. (2018) gör dock inte samma tydliga koppling mellan specifika ämnen och CT. De likställer istället i hög grad undervisning i programmering med undervisning i CT. 287 elever i årskurs fyra och fem svarade på en onlineenkät om deras attityd till programmering. Resultaten visar på hur meningsfullt och intressant elever tycker att det är att lära sig programmering. Kong et al. (2018) förslår utifrån sina resultat att undervisning i programmering bör designas för att passa elevernas intresse vilket då även skulle gynna elevernas CT.

Eftersom inte alla elever ska eller kan bli programmerare är Saari et al. (2016) forskning av intresse då de undersöker grundskollärares förståelse och motivation för att både själva lära sig och att undervisa i programmering och CT. I studien ingick både erfarna lärare och lärarstudenter. De intervjuades före och efter de fått en-till-en undervisning i grundläggande programmering. I den första intervjun var det flera av lärarna som uttryckte både förvirring och till viss del rädsla då de var osäkra på vad programmering innebar. Vid det andra intervjutillfället när de fått pröva själva hade känslan av förvirring minskat och ingen av läraren uttryckte längre någon rädsla. Lärarna som deltog i studien blev säkrare i att hantera den digitala utrustningen och blev mer

förtrogna i CT vilket Saari et al. (2016) menar även kommer att påverka deras förmåga att undervisa i programmering.

I en svensk undersökning visar Nouri et al. (2020) vad lärare som arbetat med programmering i skolan några år (redan innan det blev obligatoriskt) uppfattar att eleverna får möjlighet att lära sig av undervisningen. Den tematiska analysen av lärarnas svar visar att många av de förmågor eleverna använder när de programmerar är CT förmågor. Men det handlar samtidigt om mer än kodning- tankearbetet som krävs är större än så och kan vara viktiga även i andra sammanhang. Utöver rent praktiska datorkunskaper lyfter lärarna hos Nouri et al. (2020) fram allmänna förmågor såsom problemlösning och samarbetsförmågor. Samtidigt ser de även att eleverna tränar sina språkliga förmågor. Det innebär att lärarna uppfattar att eleverna använder ett antal förmågor vilka alla är essentiella för lärande och digital kompetens.

3.4 Engagemang för att lära programmering

Med ett designorienterat perspektiv sätts fokus på vilka möjligheter och begränsningar det finns för att kunna engagera sig i den alltmer multimodala verkligheten (Kress, 2000). Engagemang, både angående vilka möjligheter elever har att delta i inlärningsprocessen och på vilket sätt engagemanget uttrycks, är därför av stort intresse. Flera olika studier visar hur engagemang och resultat i undervisningen korrelerar (ex. Cochran et al., 2019; Melander Bowden, 2019). Genom att lärare och andra som undervisar blir mer medvetna om vad som kan stärka elevers engagemang, kan de även forma undervisningen på ett sådant sätt att den gynnar lärande för fler (Cochran et al., 2019). Två områden tycks särskilt viktiga för engagemang att lära sig programmering, de digitala verktygen och möjligheten att samarbeta.

För att stödja elevers engagemang kan det vara viktigt med verktyg för programmeringsundervisningen som passar målgruppen. Asad et al. (2016) undersökte grundskoleelevers attityd till att lära sig programmering. 24 elever i årskurs fyra och fem undervisades under tre veckor i en specialdesignad kurs med stöd av logiska spel, blockprogrammering och textprogrammering. För- och eftertest med enkäter över elevernas uppfattning kring programmering genomfördes vilka kompletterades med kvalitativa data i form av observationer när eleverna arbetade. Resultaten visar att elevernas grundläggande attityd var

fortsatt positiv efter genomförd kurs och att de samtidigt förbättrade sin problemlösningsförmåga. Asad et al. (2016) förslår därför att det är lämpligt att undervisa med stöd av interaktiva digitala miljöer i grundskolan. Även Ouahbi et al. (2015) har undersökt elevers uppfattning om programmering. Studien genomfördes experimentellt genom att testa vilka programspråk som motiverar elever på motsvarande svensk gymnasienivå att lära sig programmering. De två programvaror som användes var Pascal (textbaserat) och Scratch. Resultaten visar på att eleverna som använde Scratch uppfattade det mer motiverande att lära sig programmering i jämförelse med de som undervisades mer traditionellt. Dessutom var det en högre andel av eleverna i Scratch-gruppen som ville fortsätta med programmering efter projektet.

Samtidigt som de redovisade undersökningarna talar för visuellt blockbaserat programmeringsspråk kan det finnas andra aspekter som talar för andra sätt att undervisa och att en kombination av olika programspråk kan vara positivt. Chang et al. (2017) undersökte vilken typ av verktyg som kan stötta elevers intresse och motivation för att lära sig programmering. De undersökte hur visuellt programmeringsspråk, jämfört med traditionellt textbaserade användargränssnitt påverkar elevernas engagemang. Studien genomfördes i tre steg. I det första steget undersöktes först alla deltagarnas grundläggande motivation för att lära sig programmering. I det andra steget undervisades deltagarna först i traditionell textbaserad programmering och därefter i visuell programmering. I det tredje steget undervisades deltagarna istället först med visuell programmering och efter det i textbaserad programmering. Därefter testades deltagarnas motivation igen. Resultaten visar på att det är en positiv väg att kombinera visuell och textbaserad programmering och att då börja med den visuella programmeringen. Det beror på att komplexiteten att lära sig gränssnittet minskade och istället för att memorera olika kommandon kunde deltagarna koncentrera sig på att lösa problem. När de lärt sig grunderna var det lättare att sedan överföra kunskaperna i den textbaserade miljön. Chang et al. (2017) menar att detta gav eleverna engagemang att gå vidare och lära sig mer om hur programmering fungerar.

Genom att utgå från att elevernas möjlighet att delta i undervisning beror på deras engagemang är den sociala omgivningen viktig. Vardagserfarenheter som används i lärprocessen kan då ge stöd för ökat engagemang. Melander Bowden (2019) undersökte bland annat elevers engagemang i samband med digitala aktiviteter. Studien bygger på

videoobservationer av två elever i årskurs fyra som i ämnet bild löser programmeringsuppgifter tillsammans. Med stöd av sin lärare, kunde eleverna använda sina erfarenheter av olika datorspel i hemmiljö till att slutföra sina uppgifter i programmering. Elever som använder digital utrustningen hemma och sedan lär sig att programmera i skolan har möjlighet att bli både konsumenter och producenter av digitala aktiviteter. Även om det är viktigt att beakta att det finns elever som inte har erfarenhet av vare sig programmering eller datorspel hemifrån, visar resultaten hur programmering kan vara ett bra sätt att integrera barns vardagserfarenheter in i undervisningen (Melander Bowden, 2019).

Engagemang kan även innebära att samspela med kamrater och dela med sig av sina kunskaper. Harlow och Leak (2014) har kartlagt hur olika idéer för problemlösning vid programmering sprids i ett klassrum. 20 elever i årskurs tre deltog i studien, alla lektioner videofilmades och analyserades därefter i flera steg. I ett första steg analyserades sekvenser där eleverna pratade om olika lösningsförslag, med vilka de pratade samt om de berättade om sin lösning för någon vuxen. Därefter gick man tillbaka och analyserade hur dessa idéer kom till och hur de utvecklades vidare. Till sist klustrade man de olika lösningsförslagen och idéerna för att se hur de hörde ihop. Resultaten visar att när eleverna delade med sig av sin erfarenheter utvecklades deras problemlösningsförmåga i matematik och lösningarna blev mer avancerade (Harlow & Leak, 2014). För att dela med sig av sina idéer var det viktigt för eleverna att få uppmärksamhet, både av kamraterna och av de vuxna. Det gav energi till att engagera sig och berätta om sina lösningar.

Just att kunna samarbeta och lösa problem tillsammans är viktiga egenskaper inom programmering (Lewis, 2011). För ett gynnsamt samarbete där alla elever har möjlighet att bli engagerade, är det viktigt att ha tydlig rollfördelning och att gruppsammansättningen är väl genomtänkt. Detta är slutsatser som Lewis (2011) drar efter att ha jämfört olika sätt att organisera elevers möjlighet att samarbeta vid programmeringsundervisning. Två olika undervisningsdesigner användes, att två elever samarbetar vid samma dator respektive att två elever samarbetar men har varsin dator. De mest positiva resultaten gällande elevers attityd till programmering var hos de elever som hade sin egen dator. Då slapp de konflikter om vem som skulle sköta datorn och varje elev hade möjlighet att känna att de hade tillgång till att lösa programmeringsuppgifter på ett sätt de fann tillfredsställande. Detta ledde bland annat till minskad frustration. Samtidigt fanns fördelar med att

samarbeta och lära av varandra. Lewis (2011) menar att fortsatt utveckling av att programmera tillsammans i par är något som kan gynna elevernas engagemang.

Även Tsan et al. (2018) drar slutsatsen att ett framgångsrikt samarbete i programmering kännetecknas av att eleverna är engagerade i meningsfulla diskussioner där alla elever bidrar med idéer till lösningar. I deras studie undersöktes just hur elever engagerar sig när de programmerar. Datamaterialet samlades in under ett år då ca 55 elever på en grundskola i USA deltog i frivillig en programmeringskurs. Lektionerna videofilmades och fem par elever valdes ut till analysen och resultaten redovisas i form av en fallstudie. Analysen koncentrerades på hur eleverna delade med sig av sina egna, och gav respons på varandras förslag att lösa olika uppgifter. Eleverna som arbetade tillsammans gav flera olika typer av förslag på lösning, ibland genomfördes förslagen men det fanns även gånger då förslagen från kamraterna helt ignorerades. Både bra och mindre bra förslag på lösningar användes eller förkastades utan att eleverna diskuterade konsekvenserna. Det kan bero på att eleverna kanske inte förstod vad förslagen innebar. Tsan et al. (2018) menar därför att man bör uppmuntra till dialog genom att designa för en undervisning där alla har möjlighet att komma till tals. Då kan elevernas samarbete utvecklas och leda till lösningar där alla elever i gruppen är engagerade.

3.5 Syntes

Denna sammanställning av tidigare forskning visar på att det är svårt att lyfta fram en metod eller ett sätt som kan anses vara rätt väg att gå för att barn och unga ska lära sig programmera. Det finns bland annat ett flertal studier som fokuserat på att jämföra olika sorters kod. Ofta visar resultaten mer på elevernas attityd till programmering än på vad de har möjlighet att lära sig. Resultaten är inte entydiga, det som kan anses vara bäst i ett sammanhang fungerar inte lika bra i andra.

Synen på hur datalogiskt tänkande (CT) definieras varierar. Det finns dock en samsyn om att använda CT inte är det samma som att kunna programmera men att CT kan appliceras på undervisning i programmering. I vissa fall har man undersökt förhållandet mellan CT och undervisning i andra ämnen. Det är dock svårt att visa på transfereffekter då det är svårt att klarlägga alla orsakssamband.

Det finns även forskning kring programmeringsundervisning som fokuserar på hur elever har möjlighet att vara engagerade i sitt lärande. Möjligheter till engagemang kopplas ofta till lärmiljön och med tillfällen att samarbeta med kamrater och att utbyta erfarenheter.

Med utgångspunkt att denna studie har ett elevperspektiv och fokuserar på elevers lärprocesser ger den ett intressant bidrag till forskning om hur elever designar för och i sitt lärande i programmering.

4. TEORETISKA PERSPEKTIV

I detta kapitel redogörs för det teoretiska ramverket som ligger till grund för denna studie, *Design för lärande* (Leijon & Lindstrand, 2012; Kress & Selander, 2012; Selander, 2008; 2017). Ramverket används för att utforska elevers lärprocess och bidrar även till diskussionen om hur man kan förstå kopplingen mellan lärande och design. Lärande sker över tid och i olika situationer, det kan beskrivas som att lärande sker i sekvenser. Med den utgångspunkten används modellen *Learning Design Sequences* (LDS) (Selander, 2008, 2017) som analysredskap. Hur engagemang påverkar de val som styr designen under en lärprocess är en viktig aspekt i ett designperspektiv (Kress & Selander, 2012). I kapitlets andra avsnitt presenteras hur elevers engagemang kan definieras och identifieras.

4.1 Designorienterad teori

Selander (2017) lyfter fram att ett designorienterat perspektiv sprunget ur tidigare sociokulturella teorier kan bidra till diskussioner om hur nya meningsfulla sammanhang för lärande kan skapas. Genom att se på lärande i ett socialsemiotiskt designperspektiv finns det möjlighet att se lärandeprocesser ur perspektiv som kan inkludera flera typer av lärsituationer. Lärande blir därför inte enbart ett härmande eller kopierande av andra utan sker i olika sammanhang och på olika sätt. Det skiljer sig exempelvis vilka semiotiska resurser du använder om du ska tolka en bild på bildlektionen eller har möjlighet att agera med hela kroppen i ett socialt sammanhang under en dramalektion (Kress, 2000). Lärande blir till en process som inkluderar olika sätt att delta, pröva och gestalta (Selander & Kress, 2017).

4.1.1 Informellt och formellt lärande

Vi lever i en värld som är multimodal (Kress, 2015). Det vill säga att vi kommunicerar på många olika sätt och agerar i olika sociala sammanhang på ett helt annat sätt än vad som var fallet för bara några årtionden sedan. Utvecklingen går snabbt, därför måste även synen på vad som är lärande vara i konstant rörelse. Lärmiljöer och lärresurser är något som ständigt är i förändring. Det är därför sällan värdefullt att diskutera att ett sätt att lära är bättre än det andra (Kress, 2015). Det som förr kanske räknades som ett lärande rum det vill säga klassrummet på en skola är numera ett betydligt mer vidgat rum där de informella miljöerna och digitala medierna allt mer

räknas in. Selander (2017) menar att det inte finns någon skillnad mellan lärande i olika situationer, samma komponenter ingår vare sig det lärandet sker på fritiden eller i skolan. Det är dock skillnad mellan olika inramningar, det vill säga om det är en formell, semiformell eller informell miljö.

Informellt respektive formellt lärande, kan definieras med avseende på huruvida de erbjudna aktiviteterna är obligatoriska eller ej (Sasson, 2014). Informellt lärande kan ske överallt, när som helst (Dawson, 2014). Det är oftast inte uppstyrt utan sker exempelvis i olika typer av fritidsaktiviteter och andra vardagliga händelser. Lektioner i skolan kan däremot betraktas som formella lärandeaktiviteter. Under ideala förhållanden finns ett mål med vad som ska uppnås med lektionen och den har planerats med ett tydligt syfte som grundats i styrdokument. Dawson (2014) lyfter fram miljön på SC och museer som något däremellan, miljön är designad och institutionellt inramad och det finns dessutom personal som kan vägleda besökarna. Miljön kan därför betraktas vara semiformell. Det vill säga, verksamheten är utformad för lärande, men det inte obligatoriskt att delta (Selander & Kress, 2017). När skolor utnyttjar SC för en del av den obligatoriska utbildningen kan det således finnas en spänning i vilken typ av lärande som avses. Den kreativa miljön på SC kan ge inspiration att lära sig nya saker man inte möter i klassrummet, samtidigt kan de formella normen från skolan begränsa vad som räknas som kunskap.

I denna studie är inramningen för lärandet semiformell på gränsen till formell i och med att datan är insamlad under lektioner för grundskoleelever som besöker SC under en del av sin skoldag.

4.1.2 Design för lärande

Design för lärande ger en möjlighet att underlätta förståelsen av lärande i olika miljöer och situationer (Leijon & Lindstrand, 2012). I ett designperspektiv innefattas lärande av olika sorters kunskap, det vill säga att lärande inte enbart står i relation till i förväg uppgjorda mål angivna i kursplaner eller andra styrdokument (Selander & Kress, 2017). Alla situationer kan alltså bli en lärandesituation, ibland planerad, ibland spontan. Att lära sig innebär att hela tiden bearbeta och re-designa sitt eget lärande med hjälp av de val man gör i olika situationer. Den lärande är inte passiv mottagare av kunskap utan är själv med och formar och påverkar med sitt sätt att agera förutsättningarna för förståelse och ny kunskap.

Design som förhållningssätt inom didaktik sätter i första hand fokus på förändring och utveckling (Insulander et al., 2017). Design inkluderar då både kultur och artefakter såväl som sociala egenskaper hos den som designar (Forsling, 2019). Lärandedesign handlar inte om att designa färdiga produkter utan kan vara sätt att lösa problem och ett sätt att skapa mening (Insulander et al., 2017). Design kan vidare ses både som produkt och process (Levinsen & Sørensen, 2019). Designen kan göras för andra personers lärande (produkt), men kan samtidigt vara pågående då var och en designar i sitt eget lärande (process). Design i lärandesammanhang, didaktisk design, har alltså många olika funktioner. Selander och Kress (2017) delar in den didaktiska designen i tre delar, Design *för* lärande, Design *i* lärande samt Erkännandekulturer.

I pedagogiska sammanhang ger designen och iscensättningen som läraren planerat struktur och ordning inför en lärsituation. Dessa förutsättningar skapar möjligheter för att elevernas kunskap ska kunna transformeras och framställas på nya sätt (Forsling, 2019). Med andra ord, de yttre förutsättningarna och tillgången till olika slags resurser blir till en design *för* lärande. Den påverkar i sin tur hur en individ har möjlighet att designa *i* sitt eget lärande (Leijon & Lindstrand, 2012). Lärare och elever kan då få olika roller i olika situationer. Vilken roll som är i fokus beror på ur vilket perspektiv man betraktar designen. Exempelvis ses lärare som en designer när de genom sin planering och de resurser som finns till hands, designar *för* elevernas lärande. Det finns även tillfällen då elever designar *för* lärande genom att vara delaktiga i planeringen och iscensättningen. Samtidigt finns en annan designprocess genom att eleven designar *i* sitt eget lärande exempelvis genom att använda hela eller enbart delar av det som läraren planerat. Ett möte sker mellan den planerade designen *för* lärande och designen *i* lärande som formas allteftersom genom hela lärprocessen. Således kan både de som planerat för lärande och de som lär anses vara didaktiska designers (jfr Leijon & Lindstrand, 2012) och lärande designers (jfr Levinsen & Sørensen, 2019).

Lärande kan uttryckas i transformativa processer (Kress & Selander 2012; Selander, 2017). I en transformativ process måste den lärande börja med att tolka situationen, vad som förväntas och vilka resurser som finns att tillgå. I lärprocessen används olika resurser och allteftersom formeras lärandet i något som skulle kunna presenteras som ett resultat av denna process. Viktiga aspekter i denna transformerande lärprocess är den lärandes uppmärksamhet och engagemang gällande kunskapsinnehållet

(Leijon & Lindstrand, 2012). Vi kan inte se exakt när eller hur den lärande lär, men vi kan se spåren av lärandet (Selander, 2017).

Det är inte självklart att allt lärande faktiskt ses som just lärande. Den tredje aspekten av lärandedesign, erkännandekultur, är beroende på hur utvärdering av kunskapen planeras och genomförs. (Selander & Kress, 2017). Kunskap värderas olika beroende på vilket sammanhang den lärande befinner sig i. I relation till sammanhanget är det vissa tecken på lärande som blir viktiga, medan andra inte får någon uppmärksamhet (Leijon, 2010). Det är i denna del en av skillnaderna ligger mellan formella och informella lärmiljöer. Genom designen av exempelvis betygskriterier och utvärderingar, kan vad som erkänns som kunskap i ett formellt skolsammanhang styras i olika riktningar. Många gånger är det kunskapsinnehållet läraren väljer att ta upp i undervisningen som är mest avgörande och räknas som den riktiga, viktiga kunskapen. Även om eleverna lär sig något annat samtidigt är det inte det som erkänns som kunskap. I en lärsituation på fritiden finns sällan några formella krav på kunskap eller någon som talar om vad som är viktigt att kunna. Vill någon exempelvis lära sig att programmera på fritiden kan hen själv välja vad den vill koncentrera sig på. Det är ingen som behöver erkänna kunskapen. På SC och muséer råder traditionellt heller inga formella krav på kunskaper och centren kan designa för lärande inom vitt skilda områden. Det kan alltså skilja vilket fokus SC har i sin programmeringsundervisning med vad som är intentionen i exempelvis grundskolans läroplan.

4.1.3 Learning Design Sequences

I denna studie används begreppet design teoretiskt och analytiskt för att i första hand tolka elevernas design vid de specifika inlärningsituationerna. Fokus blir då på design som process där eleverna är med och designar hur processen utformas (Levinsen & Sørensen, 2019). För att analysera denna process används en modell som beskriver hur lärande transformeras i sekvenser, det vill säga Learning Design Sequences (LDS) (Selander, 2017). Valet av analysmodell stöds av att den är väl beprövad i olika sammanhang. LDS har tidigare använts i samband med undervisning som stöds av olika digitala teknologier såsom datorer och lärplattor (Kjällander, 2011; Åkerfeldt, 2014) och i sammanhang med olika grader av informellt och formellt lärande (Insulander et al., 2017; Selander, 2017). Skolverket visar på LDS som en modell för lärare att använda för att planera och följa upp undervisning i en av sina moduler i lärportalen för lärare (Skolverket, u.å.).

Modellen är intressant då den bland annat erbjuder en möjlighet att analysera de fysiska förutsättningarna, inklusive de läranderesurser och institutionella normer som påverkar lärprocessen (Selander, 2017). SC anses alltså i denna studie vara en semiformell miljö, på gränsen till formell och det blir då intressant att analysera hur dessa förutsättningar inverkar när eleverna ska lära sig att programmera. Samtidigt ger modellen alltså en struktur för att analysera elevernas aktiviteter, sekvenser i lärandet. Då denna studie innefattar både digitala- och semiformella miljöer prövas LDS-modellen i ett nytt sammanhang.

Figur 1. *Modell över Learning Design Sequences*. Från Selander, S. (2017). *Didaktiken efter Vygotskij. Design för lärande*.²

LDS-modellen består av tre delar: iscensättning, första transformationscykeln och andra transformationscykeln (Selander, 2017). Enligt denna modell sker lärande över tid och transformeras och formeras av eleverna i det sammanhang där de befinner sig. Processen är inte alltid linjär, de lärande kan även röra sig mellan första och andra transformationscykeln. Den första delen av modellen, *iscensättningen*, inkluderar villkoren och syftet med lärsituationen (Selander, 2017). I en formell lärsituation är lärare både som person och de didaktiska val de gör av betydelse för hur iscensättningen gestaltas. Hur lektionerna planeras blir därför en viktig del i hur elevernas lärprocess kommer att se ut. Iscensättningen kan även användas för att beskriva och analysera de potentiella resurser de lärande har att tillgå. Resurser i detta sammanhang

² Selander (2017, s.150). Används med tillstånd enligt mailkonversation den 8/9, 2020.

är inte bara materiel och digital utrustning utan även den specifika fysiska miljön. I miljön ingår även de institutionella normerna för platsen där iscensättningen äger rum. Det vill säga bland annat hur man förhåller sig till samarbete och kommunikation vilket kan skilja både mellan olika formella lärmiljöer och i förhållande till informella lärmiljöer. Även de personer som deltar i lärprocessen är en del av iscensättningen då varje person har unik förförståelse och personlig förmåga att engagera sig i sitt lärande.

Den *första transformationscykeln* används för att analysera vilka typer av aktiviteter de lärande gör i en lärsekvens (Selander, 2017). Det handlar om hur olika personer bearbetar och ger uttryck för sin förståelse på olika sätt (Insulander et al., 2017). Det vill säga vad de gör, hur de gör samt vilka resurser de faktiskt använder. Den lärande behöver tolka iscensättningen för att kunna göra sina egna insatser i samspel med omgivningen. De lärandes fortsatta engagemang och vilka val de gör i denna del av processen påverkar hur lärandet sedan gestaltas.

Nya kunskaper gestaltas i en form av representation i den *andra transformationscykeln*. Ibland i form av fysiska artefakter eller i form av att ett problem har fått en praktisk lösning. Andra gånger blir kunskapen synlig genom de lärandes reflektioner och diskussioner. Vad som räknas som kunskap kan alltså bli viktigt i denna del. Inom formellt lärande där exempelvis läroplaner styr lärandesinnehållet, kan även en slags summativ bedömning av kunskaper göras i denna cykel (Selander, 2008; 2017).

LDS är ingen fix modell där man linjärt kan analysera hur det går till när någon ska lära sig något specifikt. Lärande sker över tid och ofta i cykler där den lärande kan omformera sina kunskaper flera gånger. I föreliggande studie används därför modellen som ett stöd för att uppmärksamma olika sekvenser av elevers lärprocesser i programmering.

4.2 Engagemang

I ett designperspektiv kan lärande ses som en ökad förmåga att engagera sig i världen på ett sätt som är meningsfullt för individen (Selander & Rostvall, 2010). Samtidigt kan olika typer av lärandeaktiviteter kräva olika typer av engagemang (Kress, 2000). Den didaktiska designen i iscensättningen blir därför viktig för hur eleverna kan fokusera sitt engagemang. Det vill säga hur upplägget för undervisningen planeras med

både fokus både på innehåll och struktur. Genom att studera hur elever i olika situationer investerar sitt engagemang och riktar sin uppmärksamhet får vi ledtrådar till hur deras lärande transformeras (Leijon & Lindstrand, 2012). Engagemang kan följaktligen spela en viktig roll hur lärprocesser ser ut och hur förändras över tid. Samtidigt är det viktigt att lyfta fram att det inte enbart är en teoretisk design som påverkar hur eleverna blir engagerade. Lärare och andra behöver en förmåga att stötta eleverna att utnyttja de handlingsmöjligheter designen erbjuder (Bergdahl et al., 2018). Att samarbeta och att dela med sig av erfarenheter kan då bli viktiga pusselbitar för att elever ska bli och förbli engagerade.

4.2.1 Definition av engagemang

Jag väljer i mina analyser använda mig av begreppet engagemang. Människor formas genom hur de engagerar sig i den sociala världen men även hur generationer tidigare har engagerat sig (Kress, 2015). Kress (2015) menar vidare att det finns en möjlighet för människor att själva bestämma hur de riktar sitt engagemang, vilket bland annat kan styras av personens intressen. I den multimodala värld vi lever i så är resurserna för kommunikation både många och varierade, det vill säga inte enbart begränsade till tal och skrift. Genom att studera det observerbara engagemanget genom elevernas olika sätt att kommunicera får vi då en pusselbit i hur elever designar sin lärprocess

Motivation används ibland synonymt med engagemang men betraktas i denna studie vara två skilda begrepp. Teorier om engagemang och motivationsteori är dock nära besläktade, och teorier om motivation kan ge stöd för studier av engagemang och vice versa (Fredricks et al., 2004). Motivationsbegreppet innefattar både yttre- och inre motivationsfaktorer som även påverkar människans möjligheter att lära. Yttre påverkansfaktorer kan vara belöningar eller hot om straff och inre påverkansfaktor kan motsvaras av genuint intresse och glädje (Ryan & Deci, 2000). Enligt Fredricks och McColskey (2012) är engagemang ett synligt resultat av motivation vilken ger drivkraft för lärande (Fredricks et al., 2004).

Halverson och Grahams (2019) översyn av aktuell forskning om engagemang och lärande visar att olika forskare definierar 1–7 typer av engagemang med olika indikatorer. Det råder dock en konsensus om tre typer av engagemang som är återkommande. Dessa typer är; beteendemässigt, kognitivt samt emotionellt/ affektivt engagemang.

Tabell 1 *Indikatorer som används i studien för att identifiera de olika typerna av engagemang.*

Typ av engagemang	Indikator på engagemang
Beteende	medverkan i akademiska, sociala eller andra aktiviteter; följa regler; följa (klassrums-) normer; frånvaro av störande beteende
Emotionellt	positiva och negativa reaktioner på aktiviteter, utrustning, människor (ex. lärare eller klasskamrater) eller lärandemiljön; känslor av tillhörighet eller att vara viktig
Kognitivt	vilja att göra nödvändiga ansträngningar för att förstå komplexa idéer eller behärska svårigheter; vara strategisk; vara kreativ

Kommentar: Definition baserad på Fredricks et al. , 2004; Fredricks och McColskey, 2012.

I en skolsituation eller annan designad lärsituation innebär *beteendemässigt engagemang* exempelvis att följa regler och att följa (klassrums)normer. Det vill säga att vara tyst när andra har ordet, lyssna, sitta still på sin plats om inget annat anges, sköta tider och så vidare. Beteendemässigt engagemang innebär också att delta aktivt i både sociala och teoretiska aktiviteter. *Känslomässigt engagemang* innebär fokus på positiva och negativa reaktioner. Exempel på reaktioner är glada utrop, arga kommentarer eller besvikna suckar. I en lärsituation kan reaktionerna gälla både aktiviteter och utrustning (gillar inte programmering, roboten är söt) eller personer och deras agerande (till exempel lärare eller klasskamrater). Att känna tillhörighet eller utanförskap och att vara viktig/oviktig i ett speciellt sammanhang är andra delar av det känslomässiga engagemanget. Genom att vara kreativa och strategiska kan elever visa ett *kognitivt engagemang*. Det innebär alltså att ha en vilja anstränga sig för att lära sig mer och aktivt arbeta för att erfara mer och

förstå hur saker hänger ihop. Det kognitiva engagemanget kan även innebära att planera i förväg för att nå sina mål.

Utan att hävda att de typer av engagemang som dokumenteras här är de enda möjligheterna, används i denna studie en definition där engagemang ses som en metakonstruktion vilken inkluderar beteendemässigt, emotionellt och kognitivt engagemang (Fredricks et al., 2004; Fredricks & McColskey, 2012). En viktig aspekt till att typerna av engagemang är avgränsade är exempelvis att en elev som till det yttre kan verka känslomässigt oengagerad kan vara kognitivt engagerad i att lösa strategiskt problem. För att dra slutsatser om huruvida en elev är engagerad under en lektion eller ej bör därför alla tre typerna av engagemang beaktas.

4.2.2 Engagemang och lärande

Trots att det fortfarande finns en viss diskrepans i hur engagemang definieras använder både lärare och forskare begreppet engagemang som en prediktor för hur väl undervisningen fungerar och att elever utvecklar sitt lärande. Tidigare studier visar att engagemang och lärande har ett positivt samband (exempelvis, Cochran et al., 2017; Hymers & Newton, 2019). När elever är engagerade i sina uppgifter finns en större möjlighet till ökad förståelse och lärande. Det kan även vara av stor betydelse att engagemanget är ömsesidigt (Klem & Connell, 2004). För elever kan det vara viktigt att de upplever att lärarna är känslomässigt engagerade och bryr sig om hur det går för eleverna.

Lärare vill även ofta se direkta tecken på engagemang eftersom det uppfattas ge omedelbar bekräftelse på att undervisningen fungerar (Abrahamsson et al., 2019). Klem och Connell (2004) visar bland annat att det finns ett starkt samband mellan det som lärarna uppmärksammade som engagemang och elevernas resultat. Även om det hos lärare finns en förståelse för olika typer av engagemang finns ett behov att uppmärksamma exempelvis det kognitiva engagemangets betydelse för lärande och eventuella fortsatta studier (Abrahamsson et al., 2019). Genom att analysera vad det är som engagerar elever, och på vilket sätt de engagerar sig, kan lärare få bättre verktyg att utveckla sin undervisning än att bara följa generella råd som att ha en varierad undervisning och att ställa öppna frågor (Cochran et al., 2017). När man ska utveckla undervisningen kan det vara mer effektivt att undersöka elevernas engagemang under tiden de arbetar istället för att enbart fokusera på slutresultatet (Chang et al., 2017).

5. METOD

Studien är genomförd med en kvalitativ ansats vilket lämpar sig väl i studier i olika sociala sammanhang. Utgångspunkten för metodvalet är att få en möjlighet att tolka och förstå elevers lärandeprocess i programmering. Genom att använda både observationer och intervjuer erhöles en rik data. Detta gav en möjlighet till detaljerade beskrivningar vilka sedan kunde arrangeras för att genomföra systematiska analyser. De båda artiklarna kompletterar varandra och när resultaten sätts samman bidrar de till en vidare förståelse för elevers lärprocesser i programmering. Alla exempel som redovisas i artiklarna och i sammanfattningarna av resultaten är hämtade från lektioner när elever besöker SC som en del av sin skoldag.

Metodkapitlet är disponerat enligt följande:

Först beskrivs i avsnitt 5.1 studiens urval och genomförande av datainsamling. Därefter i avsnitt 5.2 redogörs för hur analysen genomfördes, både som helhet och varje artikel för sig. De etiska överväganden som gjordes inför studien beskrivs i avsnitt 5.3. Avslutningsvis i avsnitt 5.4 redogörs för studiens kvalitet.

5.1 Studiens genomförande

Som teoretisk referensram för att undersöka lärprocessen på SC används *Design för lärande* (Kress & Selander, 2017; Selander, 2017). Till stöd för analysen används utöver tematisk analys (Braun & Clarke, 2006), modellen Learning Design Sequences (Selander, 2017) samt Fredricks och McColskey (2012) definition av engagemang. Datainsamlingen har gjorts genom i första hand delvis deltagande observation, delvis filmad, samt informella intervjuer och samtal. Denna kombinerade metod användes för att på det viset komma så nära elevernas lärandeaktiviteter som möjligt.

5.1.1 Urval

All datainsamling innebär att man gör någon slags urvalsprocess. Urvalet gör att det alltid finns några aspekter som väger tyngre än andra och det är därför viktigt att redovisa på vilka grunder urvalet är gjort. Jag följer råden från Twining et al. (2017) om att så tydligt som möjligt beskriva mitt urval. För möjligheten till att få så relevant data som möjligt har urvalet gjorts

målstyrt (Bryman, 2016). Vid målstyrda urval är det brukligt att använda flera nivåer och detta fall används tre nivåer. De tre nivåerna är: 1. val av SC, 2. val av skolklass samt 3. val av individ.

Studien genomfördes på tre svenska science centers under våren 2017. Den första nivån för urval till studien bestod således av val av SC. Samtliga 19 center som vid tillfället var med i svenska science centerföreningen fick via mejl till ansvarig chef, frågan om intresse att vara med i studien. Av de 10 center som var positiva till att delta valdes tre centers ut. Urvalskriterierna i detta fall bestod i att centret arbetade aktivt med programmering för grundskoleelever, geografisk spridning i landet samt varierad storlek (mätt i antal besökare per år).

Den andra nivån i urvalet skedde i samråd med bokningsansvarig på de utvalda SC. Denne person föreslog ett antal skolklasser som bokat lektioner i programmering under den tidsperiod då datainsamlingen var planerad. Klasserna kontaktades av mig via mejl till läraren, samtliga kontaktade lärare och klasser svarade positivt. Urvalskriterierna på denna nivå bestod således i att tidpunkten för lektionerna var möjlig då urvalet av SC innebar långa resor för mig som observatör, att lektionsinnehållet var programmering för nybörjare samt att gruppen bestod av elever i grundskolan. Observationen vid en lektion ställdes in då enbart två elever valde att vara med i studien vid det aktuella tillfället. Därefter kontaktades ytterligare en klass för att antalet observationstillfällen skulle uppgå till det planerade antalet tio stycken. Åldern på eleverna spände från ca 7 år till ca 15 år, det vill säga årskurs 1-8 i grundskolan. Det är ett stort åldersspann, men motiverades med att programmering var planerat att införas samtidigt i alla årskurser. Det fanns därför ett intresse att studera elever i olika åldrar som inte undervisats i programmering tidigare.

Den tredje nivå i urvalet gjordes under lektionerna då delvis deltagande observation som metod gör det svårt att fokusera på flera elever samtidigt. De observerade eleverna valdes ut efterhand under lektionen. Ingen hänsyn togs då till huruvida eleverna arbetat med programmering på fritiden eller liknade, utan alla elever betraktades under datainsamlingen som nybörjare eftersom det var på det viset lektionerna var designade. Några elever i taget observerades under olika delar av lektionerna. Under och i anslutning till lektionerna intervjuades de observerade eleverna kring de aktiviteter de genomförde.

Sammantaget deltog 220 elever fördelat på tio skolklasser/grupper under lektioner som varade i ca två timmar, totalt tjugotvå timmar. Lektionerna leddes av instruktörer från respektive SC. 48 elever deltog i informella samtal/intervjuer med observatören. I varje grupp fanns det med en lärare eller assistent från skolan, i vissa fall var dessa enbart åskådare, i andra fall var de mer aktiva och stöttade eleverna i deras arbete med programmeringsuppgifterna, se tabell 2.

Tabell 2 Översikt av data

	Lektion 1	Lektion 2	Lektion 3	Lektion 4	Lektion 5	Lektion 6	Lektion 7	Lektion 8	Lektion 9	Lektion 10
SC	SC 1	SC 1	SC 2	SC 2	SC 2	SC 3	SC 3	SC 3	SC 1	SC 1
Årskurs	8	8	6	1	4	5	8	8	3	3
Antal samtal	4	5	10	4	6	0	2	2	7	8
Resurs	Bärbar dator Scratch	Bärbar dator Scratch	Bärbar dator Lego-robot	Ipad Scratch-JR	Bärbar dator Scratch	Bärbar dator Makey-Makey Lego-robot Kojo	Bärbar dator Makey-Makey Lego-robot Arduino Kojo	Bärbar dator Makey-Makey Lego-robot Arduino Kojo	Bärbar dator Lego-robot	Bärbar dator Lego-robot
Aktiv med-följande lärare	nej	nej	delvis	delvis	delvis	delvis	nej	nej	delvis	delvis

Kommentar: Datainsamling gjord under perioden mars-maj 2017.

5.1.2 Datainsamlingen

Datainsamlingen gjordes samtidigt för båda artiklarna som ingår i denna studie. Tre SC besöktes vid sammanlagt tio tillfällen under perioden mars-juni 2017. För datainsamlingen användes i huvudsak delvis deltagande observation. Gunnarson (2019) påpekar att även om man inte aktivt deltar i aktiviteterna man observerar, påverkar man det som händer. Något som jag var medveten om och därför genomfördes datainsamlingen på ett sätt för att störa så lite som möjligt. Det påverkade även möjligheterna att filma då det var elever i vissa klasser som ville vara med i studien, men inte vara med på film. Jag anpassade mig efter förutsättningarna och följde några elever i taget. Observationerna kan därför beskrivas vara ostrukturerade då inget i förväg uppgjort observationsschema användes (Bryman, 2016). Observationerna kompletterades med intervjuer och samtal med de elever som observerades.

Intervjuerna kan likställas med informella samtal med ett syfte (Loseke, 2017). Frågorna anknöt till händelser under lektionen och elevernas tankar kring programmering. Genom intervjuerna erhöles även en möjlighet att få

förklaringar och förtydliganden från deltagarna själva. Intervjuerna spelades in digitalt. Totalt antal elever som intervjuades på detta sätt uppgår till 48 elever. Att kombinera dessa metoder gjorde att, som Loseke (2017) beskriver det, det erhöles en rikare data än om enbart en metod hade använts. I en kvalitativ observation är det viktigt att erhålla en rik data (Bjørndal, 2005). Mycket av studiens värde ligger i de beskrivningarna av aktiviteterna som ägde rum under de observerade lektionerna. Empiriskt förankrade beskrivningar ger en tillfredsställande grund för den teoretiska analysen (Larsson, 2005).

Att förstå kommunikation handlar mer än om att förstå det talade språket (Jewitt, 2017). Det handlar även om gester och blickar eller vilken hållning personer har. Det kan därför vara problematiskt att dokumentera all kommunikation som sker samtidigt under en lektion. Då det är mycket som händer samtidigt behövdes ett visst urval göras även under observationerna. Inledningsvis antecknades detaljer om lektionsupplägget samt lokalens utformning. Under introduktionen av lektionerna fokuserades därefter på elevernas sätt att vara aktiva under instruktörernas genomgång, hur de satt, gester och eventuell interaktion med kamraterna. Samspelet med instruktören det vill säga både *hur* och *vad* eleverna svarade på frågor eller reagerade på det instruktören visade antecknades därför också noga. När eleverna sedan arbetade praktiskt med de olika programmeringsuppgifterna var fokus i observationerna på hur de agerade tillsammans med sina kamrater, instruktören och med den digitala utrustningen. Den verbala kommunikation under lektionerna där de observerade eleverna deltog antecknades, men andra samtal noterades inte.

Det finns flera olika semiotiska resurser, det vill säga verktyg som skapar mening i kommunikation mellan människor (Jewitt, 2017). Under kommunikation i klassrummet noterades rörelser och gester samt ansiktsuttryck och tonfall. På detta sätt fanns möjlighet att fånga tecken för sådant som eleverna inte uttryckte verbalt. Samma typ av anteckningar som gjordes under observationerna och samtalen gjordes senare baserat på filmerna.

Efter varje observation renskrevs alla observationsanteckningar och anteckningar från filmerna på dator. Även intervjuerna transkriberades löpande. I en transkription sker en form av överföring av både det verbala och icke-verbala till text som går att läsa. På sätt och vis görs redan då en

tolkning för att senare kunna analysera kommunikationen vidare (Flewitt et al., 2017). För att synliggöra de resurser som användes i kommunikationen i klassrummet och under samtalen/intervjuerna användes ett transkriptionssystem. Systemet innebar att med hjälp av parenteser markerades när eleverna betonade vad de sa på ett speciellt sätt, skrattade eller kanske ropade rakt ut. På ett likande sätt noterades även de observerade elevernas rörelser, gester och mimik. Till exempel om de gick runt i lektionssalen, hängde över bänken med slutna ögon eller hoppade och viftade med armarna som reaktion på något som hände.

5.2 Analys

Design för lärande kan användas som ett teoretiskt raster för planering och analys av undervisning och lärande. Såsom teorin används i denna studie är det för att analysera elevers lärprocesser i programmering. All data, det vill säga de renskrivna anteckningarna från observationerna och filmerna samt de transkriberade intervjuerna analyserades i ett första steg med tematisk analys. Flera teman identifierades varav totalt fyra teman valdes ut till artiklarna. Tre teman till artikel 1 (inledning, instruktioner och tillvägagångssätt) samt ett tema till artikel 2 (engagemang). Därefter analyserades temana med stöd av teorin för lärandedesign (Selander & Kress, 2017). I den första artikeln användes LDS-modellen (Selander, 2017) och i den andra artikeln kompletteras analysen av temat även av Fredricks och McColskeys (2012) definition av engagemang. I kommande avsnitt följer en beskrivning av analysarbetet.

5.2.1 Tematisk analys

Analysmetoden för båda artiklarna som utgör basen för denna studie är i grunden den samma då allt material analyserades med tematisk analys enligt Braun och Clarkes (2006) modell. Detta innebär en arbetsgång i sex steg.

1. Upprepad läsning för att lära känna materialet.
2. Inledande kodning av relevanta avsnitt som matchade forskningsfrågorna.
3. Sortering av koder - formulering av teman.
4. Valda teman förfinas med det teoretiska ramverket i åtanke
5. Teman inklusive underteman definieras, namnges och analyseras med stöd av tidigare forskning och teoretiskt ramverk

-
6. Exempel från datamaterialet väljs ut för att representera de valda teman och sammanställa artikeln.

De tre första stegen, det vill säga upprepad genomläsning för att lära känna datan, en första kodning av avsnitt som matchade forskningsintresset samt att identifiera teman genom sortering av koderna genomfördes på hela datamaterialet. Från och med steg 4 förfinades temana med teorin för lärandedesign i åtanke (Selander & Kress, 2017) samt kompletterat för artikel 2, de olika indikatorerna för engagemang (Fredricks & McColskey, 2012).

5.2.2 Analys artikel 1

I den första artikeln användes tre av de teman som identifierats i den insamlade datan. Dessa teman namngavs efter dess innehåll; *introduktion*, *instruktioner* och *tillvägagångssätt (approacher)*. Den fördjupande analysen av dessa teman genomfördes med stöd av LDS-modellen (Selander, 2017), se avsnitt 4.1.2.

Från datan togs exempel som representerade de funna och analyserade temana. Två teman kopplades till det som i LDS benämns *iscensättningen*, introduktion av lektionerna samt instruktörernas inledande instruktioner. Det vill säga att vi analyserade vilka yttre förutsättningar det fanns för elever att lära sig mer om programmering och att programmera. Relevanta områden att studera i denna del var lektionsupplägg, vilken typ av programmeringsverktyg eleverna använde, vilken typ av mjukvara som var aktuell samt hur eleverna introducerades i arbetet.

Det tredje temat, *tillvägagångssätt*, analyserades med stöd av första och andra transformationscyklerna i LDS-modellen. *Tillvägagångssätten* beskriver i första hand hur eleverna gjorde när de som nybörjare skulle ta sig an en uppgift eller problem som skulle lösas med programmering. Den första transformationscykeln gav oss verktyg att analysera de faktiska resurserna, praktiska och teoretiska, som eleverna använde när de formerade och transformerade kunskap i programmering. Den andra transformationscykeln stod i fokus när vi analyserade elevernas diskussioner och tillfällen till metareflektion. I denna del av arbetet kunde vi även analysera vilka typer av förmågor eleverna använde i respektive *tillvägagångssätt*. En inlärningsprocess är sällan linjär, transformationscyklerna kan därför överlappa varandra. Rörelsemönstret,

hur eleverna rörde sig i transformationscyklerna varierade på skilda sätt i de identifierade tillvägagångssätten.

5.2.3 Analys artikel 2

Till den andra artikeln valdes enbart ett av de initiala temana ut, elevernas engagemang. Här användes fortsättningsvis även Fredricks och McColskeys (2012) flerdimensionella definition av engagemang som utgångspunkt i analysarbetet. Ord som generellt är kopplade till alla typer av engagemang markerades i hela det utskrivna materialet, det vill säga de nedtecknade beskrivningarna av den multimodala kommunikationen och de noterade händelserna under lektionerna. Särskilt intressanta avsnitt kodades rad för rad med hjälp av indikatorerna för de olika typerna av engagemang (Fredricks & McColskey, 2012). På det här stadiet identifierades en viss typ av engagemang som mer signifikant i de analyserade exemplen. Detta användes för att benämna de olika excerpt som användes i den fortsatta analysen. Dessa excerpt analyserades för att i detalj förstå de möjliga kopplingarna mellan de olika typerna av engagemang. Till artikeln lyftes några exempel ut för att illustrera resultaten.

5.3 Metoddiskussion, etik och kvalitet

5.3.1 Etiska överväganden

Som vid alla forskning är det etiska förhållningssättet viktigt. Då syftet med detta arbete innebär att undersöka unga elevers lärlprocess under lektioner i programmering gjordes noggranna etiska överväganden. Inledningsvis var det därför viktigt att ställa frågan vilka nytta detta projekt kan ha. Projektet motiveras med att då programmering är ett relativt nytt område i svensk grundskola och skollagen föreskriver att all undervisning ska vila på vetenskaplig grund och beprövad erfarenhet (SFS 2010:800, 1 kap. 5 §) är det angeläget att i så tidigt skede som möjligt undersöka hur sådan undervisning kan gå till.

Vetenskapsrådet grundläggande kodex gör gällande att alla personer som medverkar i ett forskningsprojekt ska värnas från alla typer av skada eller kränkning (Vetenskapsrådet, 2017). Vid en analys kring dessa frågor innan studien inleddes, gjordes bedömningen att med noggranna avvägningar och hänsyn till omständigheter vid varje observation och samtal, inga risker för vare sig skada eller kränkning förelåg. I linje med detta och

Twining et al. (2017) gavs information om studien skriftligt till samtliga deltagare. Informationsbrev om syftet med studien och en planering på hur genomförandet skulle gå till, skickades till de ansvariga för varje klass (se appendix 2 och 3). I breven informerades även att deltagandet var frivilligt och kunde avbrytas även av deltagare som till en början valt att vara med. Samtycke för observation och videoinspelning inhämtades genom att blanketter för godkännande om deltagande (vilket för elever under 15 även krävs av vårdnadshavare) sändes ut tillsammans med informationsbreven. Samtycke ska kontrolleras i två led (Vetenskapsrådet, 2017) vilket skedde genom att deltagarna vid varje tillfälle även informerades och frågades muntligen innan observationerna började. Datainsamlingen anpassades därefter till att enbart gälla de elever som önskade och hade medgivande att vara med vilket säkrade upp att ingen mot sin vilja skulle vare sig filmas eller observeras.

Forskning där data samlas in med videoinspelning riskerar att inkräkta på individens integritet eftersom de skulle kunna identifieras på bild. Det är därför nogra både att de som filmas och deras vårdnadshavare (vid deltagare under 15 år) godkänner detta. I de grupper där det fanns elever som inte ville eller inte hade skriftligt tillstånd avstods därför från att filma. Vidare är det viktigt att datamaterialet förvaras säkert och med hänsyn till dataskyddsförordningen, i dagligt tal GDPR (Datainspektionen, 2019). Observationsanteckningar och filmer sparas därför lösenordskyddat på en server godkänd för ändamålet. Det insamlade rådatamaterialet är enbart tillgängligt för mig, mina handledare har tagit del av delar av filmmaterialet samt transkriptioner av data som används i artiklarna.

För att i linje med dataskyddsförordningen (Datainspektionen, 2019) skydda de personuppgifter som finns tillgängliga i datamaterialet behandlas dessa anonymiserat. Det vill säga inga enskilda personer eller skolklasser kommer att kunna identifieras vid publicering. Namn som förekommer i publicerat material är pseudonymer som jag har gett. Vissa deltagare återkommer några gånger i olika avsnitt, de har då alltid samma namn. Kön och ålder stämmer med deltagarens identitet.

I linje med god forskningssed (Vetenskapsrådet, 2017) kommer datamaterialet enbart att användas i forskningssyfte. Inga kommersiella syften ligger bakom studien.

5.3.2 Studiens kvalitet

Genom att i studien redovisa både för min egen bakgrund och förförståelse, tidigare forskning samt tydligt redovisa mitt val av teori finns en god grund för studiens kvalitet (Larsson, 2005). Samtidigt är den kvalitativa forskningstraditionen med relativt liten data begränsande (Winter, 2000). Det är dock inte studiens syfte att kunna erbjuda en kvantitativ generaliserbarhet. Ett bättre sätt att beskriva kvaliteten är därför studiens tillförlitlighet, det vill säga frågor som handlar om överförbarhet och pålitlighet (Bryman, 2016).

Datainsamlingen gjordes under våren 2017 och omfattar 10 lektioner med sammanlagt 220 elever, på tre av 19 svenska science center. Den empiriska datans omfattning kan anses vara begränsad, men samtidigt är den anpassad till licentiatstudier som bedrivs på halvfart. Därför menar jag att datans omfång har gett en möjlighet till fördjupad analys av observationer och intervjuer på ett sätt som inte varit möjligt om jag hade använt mig av mer data. Redovisningen för hur datainsamling och analys är gjorda är detaljerade och kan därför anses vara pålitliga (Bryman, 2016).

Studien har sitt ursprung ur mitt eget intresse för lärande på SC, men också i ett intresse för digitala lärverktyg. Området kan tyckas smalt, men det finns en lång tradition att skolor inom närområdet till befintliga SC utnyttjar centren som lärmiljö när de egna resurserna inte räcker till. På SC finns möjlighet att pröva (för eleverna) ny teknik och det finns personal som kan ta ordinarie lärares roll som ansvarig för undervisningen. Studien är i fråga om miljön på SC begränsad till svenska sammanhang, där SC enligt sina stadgar ska erbjuda skolor stöd i ämnen som rör naturvetenskap, teknik och matematik. Detta är inte nödvändigtvis fallet för alla SC i internationella sammanhang. Eleverna i studien är på SC som en del av sin skoldag, det är alltså obligatoriskt för dem att vara där. Deras lärare, eller annan ansvarig på skolan har planerat för att SC ska ansvara för en del av elevernas utbildning. Resultaten kan därför anses överförbara (Bryman, 2016) till att inte gälla bara som en studie av lärande på SC utan även ur en aspekt där alla elever ska ha möjlighet att lära sig programmering i grundskolan.

Observationerna genomfördes som delvis deltagande observatör (Gunnarson, 2019). Det vill säga att eleverna var medvetna om min roll som observatör. Jag rörde mig bland eleverna i lokalen och pratade med dem under lektionen, men avstod från att hjälpa dem om de behövde stöd

utan bad då istället den ansvarige instruktören gripa in. Den kvalitativa datan från observationerna och intervjuerna gjorde det möjligt att beskriva de olika tillvägagångssätt som eleverna använde. Den kvalitativa datan gav också möjlighet att analysera hur tillvägagångssätten gav eleverna möjlighet att utveckla sina förmågor under programmeringslektionerna. Vidare gav studien möjlighet till att studera hur eleverna engagerade sig under lektionerna och senare analysera vad som bidrog till eller förhindrade deras engagemang under lektionerna. Beskrivningarna av lärprocesserna är således empiriskt förankrade. Det ger grund för att analysen genomfördes på ett sätt som kan anses vara tillfredställande (Larsson, 2005).

Vid analysen av elevers lärprocess används båda transformationscyklerna i LDS-modellen, det skulle kunna ifrågasättas då lärmiljön på SC betraktas vara semiformell och Selander (2017) beskriver att den andra cykeln är mest relevant i formella lärmiljöer. Samtidigt ger det som Larsson (2005) beskriver det, ett heuristiskt värde genom att tillföra en ny aspekt av lärande i programmering. I den första artikeln beskrivs hur eleverna i första hand använde sig av den första transformationscykeln när de arbetade praktiskt, men även att eleverna särskilt inom vissa tillvägagångssätt ser ut att röra sig iterativt mellan de båda cyklerna. Användningen av LDS-modellen klargör de olika stegen i lärprocessen, även om denna tolkning endast kan täcka vissa aspekter av elevernas lärande.

I den andra artikeln analyseras i stället elevernas engagemang separat. Att begreppet engagemang används kan anses något kontroversiellt då man inom socialsemiotik och design för lärande i första hand talar om det närbesläktade begreppet intresse (se exempelvis Jewitt, 2017; Selander 2017). Kress (2015) menar att ett intresse kan styra hur man riktar sitt engagemang. Det finns därför en vinst i att studera *hur* eleverna riktar sitt engagemang. Det ger en förståelse för hur elever faktiskt engagerar sig inom olika (skol)ämnen för att tolka sin omvärld (Kress, 2000). Om enbart engagemang använts som förklaring till elevernas design i lärande hade detta inneburit begränsning, då det naturligtvis inte är det enda som påverkar hur hela lärandet designas.

Att använda designteorin på detta sätt, att analysera flera delar av elevernas lärande var för sig ger en innebördsrikedom. Samtidigt kan det uppstå en otydlighet om hur de olika delarna hör ihop. Därför är den

sammanfattande diskussionen av hela studien av stor vikt för att få en uppfattning om hela lärprocessen under de observerade lektionerna. Då programmeringsundervisning inte undersökts med stöd av designteorin på detta sätt tidigare, ger studien även ett teoritillskott. Något som stärker kvaliteten ytterligare (Larsson, 2005).

6. RESULTAT

Elevers lärprocesser i programmering står i centrum för båda artiklarna som ingår i denna licentiatstudie. Till hela studien användes samma typ datainsamling, det vill säga delvis deltagande observation och informella intervjuer/samtal med elever under lektioner i programmering på SC. Den första artikeln fokuserar på förutsättningar för lärande i programmering och elevers tillvägagångssätt när de programmerar. Den andra artikeln har istället fokus på elevernas engagemang. Tillsammans utgör artiklarna en bild av grundskoleelevers design för och i lärande i programmering.

6.1 Artikel 1:

Design for learning programming

Approaches taken by novice learners

[Design för lärande i programmering

Tillvägagångssätt som används av nybörjare]

Syftet i den första artikeln är att bidra med kunskap om hur designen för lärande ser ut när nybörjare lär sig att programmera. Resultaten beskriver hur lektionerna iscensätts för lärande i programmering och presenterar fem tillvägagångssätt som eleverna använder när de programmerar. Tillvägagångssätten ger i sin tur eleverna tillfällen att använda och utveckla olika förmågor som är viktiga i ett digitalt samhälle.

6.1.1 Förutsättningar för lärande i programmering på SC

För att undersöka förutsättningarna för lärande i programmering analyserades först det som i LDS-modellen (Selander, 2017) beskrivs som iscensättning. Till iscensättningen, det vill säga designen *för* lärande kopplades två teman från empirin, introduktionen och instruktionerna.

Analysen av iscensättningen för programmeringsundervisningen på SC sammanfattas nedan:

Det fanns personal som tog emot eleverna när de kom till SC. Medföljande lärare/skolpersonal ansvarade för att elevgruppen följde SCs rutiner men hade oftast en passiv roll gällande lärandeaktiviteterna. Instruktörerna på SC ansvarade för lektionsbesökets innehåll. Lektionerna var noggrant

förberedda gällande utrustning. Datorer och robotar var laddade och hade nödvändiga program installerade. Då det hände att tekniken ändå krånglade fanns det resurser som tog hand om det. Alla aktiviteter kunde genomföras utan förkunskaper. Det fanns en viss planering för elever som kommit längre i sin kunskapsutveckling. Uppgifterna som var förberedda var företrädesvis baserade på visuell programmering.

Under introduktionen av lektionerna diskuterade instruktörerna med eleverna om deras erfarenhet av programmering. Resultatet visar på att instruktörerna strävade efter att bedöma elevernas förförståelse genom att bland annat ställa frågor om när och hur programmering används. I exemplet nedan har instruktören frågat hur en dator fungerar.

- Alltså den gör ingenting själv, den måste alltid ha hjälp av nån som håller på med den. Som en telefon om man ska typ skriva nånting kan den inte komma på det själv.

Rut, årskurs 8

Elevernas olika svar användes som en läranderesurs, det vill säga om de förslog att programmering används i mobiltelefoner utgick instruktören i sina förklaringar från det. I vissa grupper använde instruktören egna exempel, det fungerade oftast bra men det uppstod tillfällen då eleverna inte alltid kände igen de exempel instruktörerna tog upp. Ett sådant exempel var i en grupp där instruktören tog robotgräsklippare som exempel där programmering används i vardagen. Något som då inte var vardag för barn som bor i lägenhet i ett höghusområde.

På SC användes under de observerade programmeringslektionerna, två typer av instruktioner. Det första sättet innebar att instruktören gick igenom anvisningar för uppgiften ett steg i taget.

Uppgiften består i att lära sig att göra ett enkelt klickspel. Instruktören som aldrig tidigare träffat eleverna går igenom programmets huvuddelar och visar olika typer av block som ska användas. Sedan går hen igenom steg för steg grunderna för klickspelet. Eleverna ska följa med och göra stegen samtidigt vid sina datorer.

(Utdrag ur observationsanteckningar SC1)

Vid den första typen av instruktion fick alla elever ett program som fungerade från början. De hade sedan möjlighet att utveckla programmet utifrån givna uppgifter. Det var i vissa fall svårt för instruktören att hålla

ihop gruppen då några elever var snabbare än de andra att utföra alla steg. Det andra sättet innebar att instruktören visade övergripande funktioner och sedan lät eleverna läsa instruktioner själva.

...instruktören berättar i korta ordalag steg för steg de första instruktionerna [...]Instruktören släpper efter de inledande instruktionerna eleverna att arbeta fritt utifrån de instruktioner som finns i häftet. Instruktören uppmuntrar dem att testa...

(Utdrag ur observationsanteckningar SC3)

Vid den andra typen av instruktion fanns hela tiden de skriftliga instruktionerna som resurs för att genomföra uppgifterna. Eleverna använde dock i högre utsträckning möjligheten att fråga en kamrat eller instruktören, istället för att läsa när de behövde stöd.

På LEGO-robotlektionerna fick eleverna uppdrag att lösa med hjälp av roboten. Uppdragen innebar bland annat att köra en viss sträcka eller att flytta några LEGO-bitar. Uppdragen var av olika svårighetsgrad och eleverna kunde i viss mån välja vilka uppdrag de ville göra. Eleverna hade tillgång till dator där de programmerade roboten och kunde sedan testa sitt program på golvet (eller ett stort bord). På Scratch-lektionerna var uppgiften att eleverna skulle programmera ett eget spel. Det fanns då flera möjligheter att animera spelet och designa spelet med olika avatarer (sprite), bakgrunder och ljud. Under de lektioner där KOJO, Makey Makey och Arduino användes fick eleverna korta prova-på uppdrag.

6.1.2 Transformationsprocess leder till olika tillvägagångssätt

Efter inledningen av lektionerna med både introduktion och instruktioner fick eleverna tid att själva programmera. Det vill säga att de fick tillfälle att designa i sitt eget lärande vilket ledde till de tillvägagångssätt som utgör det tredje temat som analyserades. Temat tillvägagångssätt har fem underteman som framträdde i empirin: *matematiskt*, *fel-och försök igen*, *steg-för-steg*, *rutin* samt *estetiskt*. Varje tillvägagångssätt illustreras i artikeln med utdrag från observationsanteckningar och excerpt från intervjuer med elever.

De fem tillvägagångssätten användes av flera olika grupper, oberoende av ålder hos eleverna eller på vilket sätt de inledningsvis instruerats (årskurs ett- eleverna är dock borttagna från denna analys då deras programmeringsaktivitet var så tydligt åldersanpassad för yngre elever). I den tematiska analysen och med stöd av LDS-modellen (Selander, 2017)

renodlades de olika sätten för att tydliggöra likheter och skillnader. Samtidigt är tillvägagångssätten inte fixa, eleverna kan i olika sammanhang byta tillvägagångssätt.

De olika sätten gav i sin tur att eleverna tränade olika typer av förmågor. Förmågorna är något som eleverna delvis redan hade men som de också genom det tillvägagångssätt de använde kunde fortsätta att utveckla. I LDS-modellens perspektiv (Selander, 2017) transformerade och formerade eleverna sitt lärande i programmering genom att de ur analysynpunkt rörde sig från den första- till den andra transformationscykeln.

Tabell 3 Översikt på elevers tillvägagångssätt vid problemlösning i programmering

Tillvägagångssätt	Lärprocess	Lösning av problemet
<i>Matematiskt</i>	problem löses teoretiskt	i ett stycke
<i>Fel- och försök igen</i>	problem löses praktiskt	i ett stycke
<i>Steg-för-steg</i>	problem löses med hjälp av tidigare erfarenhet	ett steg i taget
<i>Rutin</i>	problem löses praktiskt och teoretiskt genom att utnyttja rutiner	ett steg i taget
<i>Estetiskt</i>	problemlösning är underordnad det estetiska uttrycket	delvis och med fokus på estetiskt uttryck

Kommentar: Eleverna kan vid olika tillfällen använda skilda tillvägagångssätt.

De elever som använde det *matematiska* tillvägagångssättet försökte lösa hela uppgiften som ett problem som går att räkna ut. När eleverna försökte lösa problemet teoretiskt i ett stycke kan man säga att de hoppade över möjligheten att vara konkreta och använde istället sin förmåga att tänka abstrakt. Det kan exempelvis naturligtvis vara möjligt att beräkna hur många varv ett hjul ska snurra för att en robot ska köra en viss sträcka, eller analysera vilka koordinater som behöver skrivas in för att en sprite ska röra sig på skärmen. Men samtidigt blev det relativt fort avancerade beräkningar och eleverna riskerade att köra fast.

Fel-och-försök igen är ett tillvägagångssätt som till en början kan tyckas vara ostrukturerat, men där eleverna oftast kom fram till en lösning. Även här försökte eleverna ta sig an hela problemet på en gång, till exempel genom

att skriva in ett långt program som de trodde motsvarade hela den sträcka en robot skulle köra för att komma fram till målet. De testade praktiskt, det vill säga mer konkret än det matematiska tillvägagångssättet. Det hände att eleverna fick göra många ändringar i programmet innan de kunde lösa hela problemet. Samtidigt använde de sin förmåga att uthålligt testa och pröva, även när de behövde börja om gång på gång.

Genom att använda tillvägagångssättet *steg-för-steg*, såg eleverna möjligheten att dela upp problem i delar för att sedan sätta ihop det till en lösning. Det gav dem möjlighet att pröva varje steg i programmet innan de gick vidare för att lösa hela uppgiften. De arbetade alltså praktiskt och prövande, men diskuterade även de olika stegen teoretiskt. Det blev ett steg-för-steg-lärande för eleverna då de utnyttjade erfarenheten från en uppgift till nästa och utvecklade programmen de skrev till att bli allt mer komplexa. Eleverna generaliserade och utnyttjade sina erfarenheter genom att samla och modulera kunskaper och idéer för att använda vid nästa steg i programmeringen.

Den *rutinbaserade* tillvägagångssättet gav eleverna möjlighet att lösa uppgifterna ett steg i taget, precis som i tillvägagångssättet ovan. Det som skiljer det rutinbaserade sättet från steg-för-steg sättet är att eleverna inte på samma sätt utnyttjade nya idéer och lösningar från de uppgifter de genomfört till nästa. De löste istället alla uppdrag på samma sätt utifrån de instruktioner de fått inledningsvis. Detta sätt att arbeta liknar en mekanisk rutin, de automatiserade och utförde alla moment på samma sätt. Rutinen blev till en förmåga som eleverna kunde använda till att planera och genomföra uppgifterna enligt ett tillvägagångssätt som gav dem möjlighet att lösa problemet trots att de till en början var osäkra på hur de skulle göra.

Det femte tillvägagångssättet, det *estetiska*, är på sätt och vis fristående från de andra. Eleverna som använde detta sätt löste sällan hela uppdraget de ställdes inför. De fastnade ofta i moment som att välja bakgrunder till spelet de skulle programmera eller leta efter det perfekta ljudet för att illustrera en sekvens i programmet de höll på med. Det gör att man kan ifrågasätta om det verkligen är ett lämpligt tillvägagångssätt om det är andra typer av problem som i första hand ska lösas. Samtidigt aktualiserar detta hur eleverna utnyttjade sin kreativitet när fokuserade på att lösa andra problem som kan vara viktiga för en helhetsupplevelse.

Då vi i artikeln diskuterar tillvägagångssätten och de förmågor eleverna använder gör vi det i relation till förmågor som är viktiga för att lära sig att programmera. Det är inte på det viset att en elev som använder ett

tillvägagångssätt enbart har en typ av förmåga, men i varje tillvägagångssätt används någon eller några mer utmärkande förmågor. Förmågorna som eleverna använder och tränar på jämför vi (men likställer inte) med det som fått uppmärksamhet under begreppet CT. CT är alltså inte något datorer gör, utan hur människor gör och tänker (Mannila, 2017; Wing, 2006). När vi sätter CT i relation till våra resultat tydliggör det olika aspekter av vad det kan innebära att lära sig att programmera.

Inom CT är det *matematiska* sättet intressant. När det gäller olika typer av representationsformer inom matematikutveckling är det ofta en fördel att gå från det konkreta via halvkonkret, halvabstrakt till abstrakt (Heddens, 1986). En del elever behöver mycket konkreta föremål som de kan ta och klämma på då, andra kan "hoppa" över vissa steg. En aspekt av CT är just att det kräver tänkande på flera nivåer av abstraktion (Cetin & Dubinsky, 2016). Abstrakt tänkande är en viktig förmåga för att förstå komplexa sammanhang inom programmering och eleverna som använder det matematiska tillvägagångssättet tränar på att använda sin abstrakta förmåga. Det *estetiska* kreativa tillvägagångssättet där eleverna i stor utsträckning använder sig av halvkonkreta representationer kan då nästan anses vara en motsats till det abstrakta matematiska. Samtidigt är kreativitet en förmåga som är viktig för att exempelvis hitta nya lösningar, som ingen annan tänkt på, något som även lyfts fram inom CT (Mannila, 2017). Att kreativitet och programmering hör ihop visade Turkle och Papert (1990) tidigt då de identifierade framgångsrika programmerare vars arbetssätt mer liknade en konstnär, än en logikers.

Något annat som lyfts fram inom CT är individernas uthållighet (Kalelioğlu et al., 2016). Det är något som tillvägagångssättet *fel-och-försök igen* visar prov på. Det vill säga att de inte ger upp, utan försöker om och om igen, ibland med ett nytt förslag på lösning varje gång. Återigen nästan som en kontrast till detta sätt att arbeta använde eleverna med tillvägagångssättet *rutin*. De lärde sig ett sätt att lösa programmeringsuppgifterna på och följde därefter det mönstret för att lösa flera uppdrag. I vissa fall kan det därför vara viktigt att lyfta fram även mer rutinbaserat arbete som en del av CT (Kalelioğlu et al., 2016; Wing, 2006). Rutinen kan ge en trygghet i hur man kan komma fram till en lösning även om man inte förstår all komplexitet i problemen som ska lösas eller har uthålligheten att själv försöka om och om igen.

Till sist kan även förmågan att generalisera även uttryckas som en CT förmåga (Durak & Saritepeci, 2018). Eleverna som använde tillvägagångssättet *steg-för-steg* gjorde först ett bit i taget, provade och när

de såg att det fungerade behövde de inte prova samma sak igen utan kunde genom att generalisera, gå vidare och använda det de lärde sig i ett steg för att utveckla nästa.

Det finns alltså likheter mellan CT och de förmågor eleverna använde. Samtidigt är ett tillvägagångssätt inte enbart begränsat till en specifik CT förmåga utan tillvägagångssätten bidrar till olika sätt för eleverna att designa i sitt lärande i programmering.

6.1.3 Sammanfattning av resultat i artikel 1

Iscensättningen och designen på lektionerna gav eleverna frihet och möjligheter att använda olika tillvägagångssätt. Tillvägagångssätten kan var och ett ge grund för lärande i programmering då eleverna även som nybörjare använder antal förmågor som kan anses essentiella för programmering. Genom att använda de förmågor eleverna redan hade innebär att de arbetade rekursivt vilket kan leda till vidare och djupare utveckling av förmågorna. Detta är i linje med vad Wing (2006) uttrycker om just CT.

Analysen visar att eleverna samarbetade när de löste programmeringsuppgifterna. Dock saknades tillfälle för gemensam reflektion, något som alltså är en del av den andra transformationscykel i LDS-modellen (Selander, 2017). Reflektion och tillfällen att lära av andras erfarenheter kan vara en viktig del i en meningsskapande process.

Studien gjorde det alltså möjligt att beskriva och analysera de tillvägagångssätt eleverna praktiserade och hur de på detta sätt använde och tränade specifika förmågor i samband med programmering. Att förstå de tillvägagångssätt som beskrivits i denna studie kan vara användbart för fortsatt utveckling när alla elever i grundskolan ska ha möjlighet att lära sig grunderna i programmering.

6.2 Artikel 2:

“I am magic!”

Pupils' engagement when designing in learning programming

[**“Jag är magisk!”** Elevers engagemang när de designar lärande i programmering]

I artikel två är det elevernas design i lärande som är av intresse. Elever designar i hög grad själva sitt lärande genom sättet de samspelar med andra och de digitala resurserna. Det ger också att de engagerar sig på olika sätt. Engagemang är flerdimensionellt (Fredricks et al., 2004) och olika typer av engagemang blir synligt i elevernas design i lärande. Engagemang är situerat, det vill säga alltid knutet till det aktuella sammanhanget (Fredricks & McColskey, 2012). Sammanhanget är i första hand programmeringsundervisning för grundskoleelever och kan därför anses vara relevant för lärande på SC såväl som i skolan.

6.2.1 Förutsättningar för engagemang

Observationerna av lektionerna i programmering på SC visar att eleverna är engagerade, särskilt om man använder begreppet engagemang i en allmän, vardaglig bemärkelse. Flera elever visar sitt engagemang med hela kroppen och spontana utrop. Samtidigt ger de ”allmänna bemärkelsen” en begränsad bild av engagemanget då vissa dimensioner av elevers engagemang ofta inte är lika synligt. Engagemang kan uttryckas både i beteendemässigt, känslomässigt samt kognitivt engagemang (Fredricks & McColskey, 2012). När dessa tre dimensioner av engagemang används vid analysen blir det tydligt att engagemang kan uttryckas på olika sätt. Att man inte är exalterad, hoppar upp och ner eller ropar rakt ut när man lyckas med en uppgift, behöver inte betyda att man inte är engagerad i uppgiften. Hur eleverna har möjlighet att engagera sig kan bero både på deras egna förutsättningar gällande personlighet och på deras tidigare erfarenheter. Vissa elever sitter tysta lutade över datorn - djupt koncentrerade, andra går runt och visar gärna sina kamrater hur de gjort sitt program. Samtidigt är möjligheten att bli engagerad beroende på sammanhanget, hur lektionerna är uppbyggda och vilken utrustning som finns till hands.

6.2.2 Exempel för att belysa olika typer av engagemang

I artikel två används tre exempel för att belysa hur engagemang tar olika uttryck när elever i samma typ av kontext deltar i programmeringslektioner för nybörjare. Anna som finns med i inledningen av denna uppsats kan lätt identifieras som att hon är emotionellt engagerad. Men det uppstår en mer komplex bild än så när analysen även innefattar elevens beteende- och kognitiva engagemang.

Det mest karakteristiska i "exemplet Anna" är det känslomässiga engagemanget. Anna uttrycker många känslor, allt från att hon tror att hon "kommer att dö" till att senare ropar ut att hon är magisk när hon förstår hur saker hänger ihop. Samtidigt är Anna kognitiv engagerad vilket även stöds av hennes positivt engagerande beteende. Hon kommunicerar mycket och diskuterar med sin kamrat hur de ska lösa uppgifterna. Anna ställer även frågor till instruktören. Hon anstränger sig för att komma på hur programmet fungerar och försöker hinna med alla uppgifter trots att tiden är begränsad. Anna fokuserar på att ta reda på hur den digitala utrustningen fungerar. Hennes önskan att lyckas är tydlig, hon är alltså inte bara känslomässigt engagerad, det är även en drivkraft genom ett kognitivt och beteendemässigt engagemang som gör att Anna förstår och kan utbrista i "I am magic!"

Det är viktigt att "se" hela engagemanget för att förstå de processer som styr elevers egen design av sitt lärande. En av slutsatserna som dras i denna delstudie är alltså att det inte bara de som syns eller hörs mest som är engagerade. På samma sätt som för Anna analyseras i artikeln därför två andra elevers engagemang, Frida och Jens. Nedan följer en sammanfattning av dessa analyser.

Frida går mest omkring eller hänger över sin stol, till synes helt beteendemässigt oengagerad. Plötsligt tänder det till och hon ger blixtnabbt sin kamrat instruktioner hur en programmeringssekvens måste definieras. När hon får frågan om hur hon tycker att lektionen i programmering är kommer en del av förklaringen till hennes något svårfångade engagemang. Hon tycker helt enkelt att det är tråkigt för att det är för enkelt. Hon har förmåga och ett kognitivt engagemang och som gör att hon mycket snabbt, trots att det är första gången som hon programmerar lego-robotar förstått hur programmeringen går till. Hon blir inte utmanad och det kognitiva engagemanget falnar. Hon hjälper sin

kamrat men har samtidigt inte tillräckligt starkt positivt beteendengagemang som gör att hon driver sitt eget lärande framåt.

Jens däremot har ett starkt kognitivt engagemang, till och med så starkt att han redan i förväg designat för sitt lärande genom att ta med sin egen datorutrustning. Han gör snabbt, för att inte bli ifrågasatt, uppgifterna som presenteras av SC instruktören, men ägnar sig sedan helt åt sitt eget projekt. Jens kognitiva engagemang för programmering smittar av sig på hans känslomässiga engagemang som är positivt för den typen av programmering han arbetar med i sitt eget projekt (Python- det vill säga programmering med textbaserad kod) samtidigt som han visar negativt känslomässigt engagemang för den visuella programmeringen hans kamrater arbetar med under lektionen.

6.2.3 Sammanfattning av resultat i artikel 2

Studiens resultat ger en möjlighet att öka medvetenheten om hur elever designar i sitt lärande kan relateras till flera dimensioner av engagemang, inte bara de typer som syns och hörs mest.

Genom observationer och samtal med elever visar studien på att beteendemässigt-, känslomässigt- och kognitivt engagemang (Fredricks och McColskey (2012) kan identifieras var för sig. De olika typerna av engagemang kan i vissa fall även vara sammanflätade. Exempelvis kan elever bli uttråkade och sluta vara aktiva om de inte utmanas kognitivt, eller så kan elevers vilja att anstränga sig leda till positiva känslor när de lyckas. Hur elevernas lärprocess designas kan alltså relateras till deras typ av engagemang. Elever som är engagerade i sin lärprocess på flera plan har möjlighet att aktivt designa sitt lärande.

7. DISKUSSION

Studiens syfte är att bidra med kunskap om hur elever designar sitt lärande i programmering. Alla deltagarna i studien var elever som var på SC under en del av sin skoldag. På detta sätt kan studiens resultat; om grundskoleelevers tillvägagångssätt när de programmerar och hur olika typer av engagemang kan relateras till elevernas läroprocess, vara relevanta även för lärande i grundskolan. I detta avslutande kapitel diskuteras resultaten i relation till uppsatsens syfte, teori och tidigare forskning.

Diskussionen är disponerad på följande sätt:

Kapitlet inleds i avsnitt 7.1 med en diskussion om elevers läroprocesser i programmering. Avsnitt 7.2 fortsätter med ett resonemang kring hur det teoretiska ramverket design för lärande används i studien. I avsnitt 7.3 sammanfattas studiens bidrag. Slutligen ges förslag på fortsatt forskning i avsnitt 7.4.

7.1 Läroprocesser i programmering

Det kan skilja om man ser design för lärande som en färdig produkt eller om designen är en process (Levinsen & Sørensen, 2019). Denna studie fokuserar på design som process och det är elevernas läroprocess i programmering som är av intresse. När designen ses som en process blir det inte något någon har gjort färdig för andra att använda, utan en design som utvecklas av både den som från början iscensätter lärsituationen och de som deltar. Studiens resultat är alltså inte förknippat med vilken typ av programmering som är bäst eller vad eleverna lär sig. Istället diskuteras här i första hand hur elever är designers i sin läroprocess när de programmerar.

7.1.1 Elever som designers

Hur väl en lektion eller varför inte en utställning på SC än är designad för lärande är designen inte färdig. Det är i mötet mellan designen *för* lärande och eleverna som elevernas egna designande börjar. Eleverna blir till designers för och i sin läroprocess genom sina erfarenheter, sina grundläggande förutsättningar och sitt sätt att vara. Utifrån LDS-modellen (Selander, 2008, 2017) se bild avsnitt 4.1.2. och den designorienterade teorin börjar alltså en persons läroprocess redan i iscensättningen. Om eleven genom iscensättningen har möjlighet att på något sätt engagera sig,

beteendemässigt, emotionellt eller kognitivt (Fredricks & McColskey, 2012) leder detta vidare genom transformationscyklerna. Här designar eleverna genom de tillvägagångssätt de använder (resultaten i artikel 1) och hur de riktar sitt engagemang (resultaten i artikel 2). I den andra transformationscykeln blir även elevernas kunskapssyn viktig, hur de ser på programmering kan vara avgörande för hur de har möjlighet att engagera sig.

Genom att på detta sätt förstå att elevernas design både *för* och *i* lärande gör dem till designers, kan vi förstå mer om vad som krävs för att de ska uppnå en digital kompetens inom området programmering. Detta sätt att se på elevernas roll i designprocessen skiljer sig något från Levinsen och Sørensen (2019) som benämner eleverna *learning designers* (lärandedesigners). Då denna studie är gjord med ett elevperspektiv är det tydligt att eleverna kan anses vara fullvärdiga designers för sin egen läroprocess.

Designer med avseende på Design för lärande

Det finns mycket som påverkar hur iscensättningen, det vill säga designen för lärande i programmering planeras. Pedagogisk kompetens, kompetens i ämnet (programmering) och teknisk kompetens är grundläggande för de lärare eller instruktörer som lägger upp planeringen (Pareto & Willermark, 2019). Det är samtidigt en mångsidig utmaning att utöver dessa grundläggande kompetenser, även möta elevers engagemang och följa upp deras tillvägagångssätt (jmf Selander, 2007).

Undervisningsmiljön är en del av iscensättningen. I denna studie är lärmiljön SC. Tidigare forskning har visat på att det är en stimulerade miljö (Falk & Needham, 2011; Shaby et al., 2017) men att man kan behöva arbeta mer med för- och efterarbete för att besöken ska bli så positiva som möjligt (DeWitt & Osborne, 2007). Här menar jag att även eleverna som besöker SC blir en tydlig del av iscensättningen. De kommer från olika skolor, de har olika erfarenheter (av programmering), de är inte där av "fri vilja" utan är på SC som en del av sin skoldag. Det blir en ny design *för* lärande med varje grupp som är på SC för att programmera.

Även valet av programmeringsspråk är en del iscensättningen, designen *för* lärande. Vilken typ av programmeringsspråk som är bäst att använda för nybörjare är något som undersökt i flera sammanhang (exempelvis; Chang

et al., 2017; Lykke et al., 2015). Det är dock svårt att säga vad som är bäst, olika programspråk har olika för- och nackdelar. Det som kan upplevas roligt, det vill säga emotionellt positivt engagerade kan samtidigt generera en osäkerhet att man lära sig rätt saker och att elevernas kognitiva engagemang inte stimuleras. Elevernas grundläggande möjlighet att använda programmeringsspråket blir en del av designen för lärande.

Sammantaget gör detta att eleverna är med och designar *för* sitt lärande i iscensättningen.

Designer med avseende på Design i lärande

Samarbete och möjlighet att interagera med både den digitala utrustningen och varandra är viktiga aspekter för lärande (Lewis, 2011; Tsan et al., 2018). När eleverna hanterar den digitala utrustningen och löser problem och uppgifter befinner sig eleverna i LDS-modellens (Selander, 2008, 2017) transformationscykler. Ofta arbetar de tillsammans och det är här som vi i första hand kan konstatera att eleverna designar *i* sitt lärande. När eleverna arbetar med programmeringsuppgifterna utvecklar de förmågor som de delvis redan har. Även elevernas flerdimensionella engagemang (Fredricks & McColskey, 2012) kan variera i den första transformationscykeln allteftersom de designar *i* sitt lärande. Det innebär att engagemanget kan variera mellan de olika typerna av engagemang och att det kan öka eller minska.

Det är framförallt i LDS-modellens andra transformationscykel som eleverna prövar om tillvägagångssätten fungerar. Vissa elever har nöjer sig med att konstatera att det fungerar och fortsätter på samma sätt. Andra elever går tillbaka och förfinar och förbättrar. När det inte fungerar som förväntat finns en risk att exempelvis elevernas emotionella och beteendemässiga engagemang förändras negativt. Hos andra elever är det kognitiva engagemanget starkt och de fortsätter därför att söka nya lösningar.

Genom att diskutera och reflektera över olika tillvägagångssätt tillsammans med andra skulle det även finnas fortsatt möjlighet för eleverna att designa *i* sitt lärande och påverka fortsättningen av läroprocessen.

Designer med avseende på erkännandekultur

Eleverna är på SC och programmerar under en del av sin skoldag. Men varken vid tidpunkten för datainsamlingen eller i dagsläget när detta skrivs (september, 2020) finns det några angivna kunskapskrav för programmering i grundskolans läroplan (Skolverket, 2019a). Samtidigt kan det diskuterats vilken typ av programmeringsspråk som räknas till "riktig" programmering (Downey, 2015; Ouahbi et al., 2015) eller om CT ska användas som begrepp vid undervisning i programmering (Mannila, 2017; Åkerfeldt et al., 2018). Resultaten i denna studie visar att vad eleverna anser som riktig programmering och värt att lära sig mer om, kan påverka hur de riktar sitt engagemang. Eleverna kan därför vara delaktiga i att designa *för* och *i* sitt lärande även i denna del av den andra transformationscykeln.

Sammantaget har eleverna därmed möjlighet att vara designers i varje del av sitt lärande.

7.1.2 Iscensättning, möjligheter och dilemman

Lärande sker i olika miljöer och sammanhang, i skolan likaväl som på fritiden. Lektionerna som studeras i denna studie har alla ett kunskapsinnehåll som handlar om att lära sig grunderna för programmering. Lektionerna är på en plats utanför skolan, men eleverna är där som en del av sin skoldag. I samband med detta finns både möjligheter och utmaningar när iscensättningen för lärandet planeras. Det finns flera studier som visar att besökare på SC ofta får en positiv bild av naturvetenskap och teknik (Dawson, 2014; Falk & Needham, 2011; Pedretti, 2004). Samtidigt som lärmiljön på SC i första hand är anpassad för ett informellt lärande föreslår Dawson (2014) att informella inlärningsmiljöer bör utnyttjas mer även i formella sammanhang då de ger en bra utgångspunkt för naturvetenskap- och teknikutbildning. Det är detta som utnyttjades vid de observerade tillfällena. Resultaten i denna studie visar på olika aspekter att beakta när man flyttar undervisningen från skolan till SC. Programmeringskunskaper hos personalen på SC och väl förbered teknisk utrustning hör till de positiva möjligheterna. Svårigheter med individanpassning och den inramade tiden för besöket hör till de dilemman som finns i detta sammanhang.

Under lektionerna på SC förväntas eleverna delta som vid en vanlig lektion i skolan. Då övergår SC från det som alltså kan benämnas en informell lärmiljö (Sasson, 2014) till en mer formell, skolliknande miljö. Det vill säga

en semiformell miljö. Selander (2017) menar att det är samma lärandeprinciper var man än befinner sig, men att synen på vad som är kunskap kan skilja. Detta gör att det kan vara svårt att etablera nya samarbeten mellan informell och formell utbildning (Kjällander et al., 2018). En anledning till det kan vara att gränsdragningen för erkännandekultur, vad som räknas som kunskap, inte går på samma ställe, se avsnitt 7.2. Då det ännu inte etablerats några kunskapskrav för grundskolan i programmering, varken vid tiden för datainsamlingen eller senare blir erkännandekulturen i detta fall något större problem. Som det ser ut idag finns det inte något tydligt sätt att bedöma programmeringskunskaper eftersom programmering i läroplanen mer beskrivs som ett verktyg för att lära andra ämnen (Nouri et al., 2020).

Det kan finnas flera anledningar att skolor ser det som en möjlighet och använder SC till en del av undervisningen. Lärares dokumenterade brist på kompetens inom programmering (Skolinspektionen 2019) kan vara en anledning. Brist på utrustning på skolan en annan. Detta kan vägas upp genom att SC:s personal förbereder lektioner i programmering och kan hantera den digitala utrustningen. Tillgången till digital utrustning på SC kan ha en potential att till viss del bidra till att utjämna den ojämna tillgången till digital utrustning på skolor som Samuelsson (2014) tidigare har beskrivit. Studien har dock inte som avsikt att ge, och ger inte heller, svar på varför de olika grupperna besöker SC.

Samtidigt observerades vissa dilemman. Att använda vardagserfarenheter som exempel i undervisningen kan vara positivt då gränsen mellan det formella och informella lärandet då suddas ut (Melander Bowden, 2019). Dock kände inte SC-instruktörerna eleverna. Det fanns därför en risk att exemplen som diskuterades under introduktionerna var något som eleverna inte har tillgång till i sin vardag och då inte motsvarade elevernas tidigare erfarenheter. Så var fallet under några av de observerade lektionerna när instruktören beskrev smarta gräsklippare eller robotdammsugare, något som inte är vardag i det höghusområde som eleverna kom från. Detta kunde i sin tur utesluta några av eleverna från viktiga delar av undervisningen.

Att möta eleverna på deras kunskapsnivå kan vara ett annat dilemma både i skolan och på SC. Det fanns elever som redan när de kom till SC hade programmerat, i olika sammanhang utanför skolan. I samma klass fanns det nybörjare som aldrig har provat programmering tidigare. Eftersom

instruktören på SC ofta är ensamt ansvarig under lektionerna kan det därför vara svårt att tillgodose allas behov.

Det finns fler dilemman, tiden är en sådan, men på olika sätt. Tiden kan verka stressande eftersom eleverna känner att de bara har en begränsad tid att lösa sina uppgifter. I vanlig skolmiljö kan man ofta fortsätta nästa lektion om man inte hunnit färdigt. Den möjligheten finns inte på SC. För andra elever är tiden istället för lång, två timmar kan vara en lång tid att hålla koncentrationen uppe. Om tiden vid SC kunde göras mer flexibel och eleverna hade möjlighet att återvända flera gånger kunde dessa problem minskas. Denna möjlighet är dock begränsad eftersom många skolor har långa resor till SC. Vanligtvis kan varje elev bara besöka ett SC några få gånger under sin tid i grundskolan. På det sätt som denna studie är genomförd går det inte att uttala sig om vad som gjordes före eller efter lektionerna. För att få ut mesta möjliga av besök på SC eller museum visar tidigare forskning att besöket kan behöva förberedas före och även följas upp i skolan efter (DeWitt & Osborne, 2007). När det utnyttjas kan dilemmat med tiden begränsas eftersom man då inte skulle behöva lägga tid på som i föreliggande fall att förklara exempelvis vad programmering är och varför det är viktigt. Sådan information skulle istället kunna finnas tillgänglig i samband med bokning.

7.1.3 Tillvägagångssätt och engagemang

När man studerar strukturen och innehållet på lektionerna i programmering i sin helhet, växlar kunskapsinnehållet under lektionerna. Från iscensättningens introduktion med ett fokus på programmering och digitalisering i samhället till att elevernas eget arbete koncentreras mer på att hantera artefakterna, det vill säga datorutrustning och robotar eller liknande. Studien visar att eleverna samarbetar och utvecklar olika tillvägagångssätt för att lösa uppgifterna. På detta sätt kan de utveckla sina förmågor i paritet med vad tidigare studier visat om programmering (Sáez-Lópes et al., 2015; Portelance et al., 2015). Resultaten gällande elevernas tillvägagångssätt visar att även nybörjare använder ett antal förmågor som kan jämföras med CT. När eleverna utsätts för rimliga utmaningar i en miljö som ger lämpliga resurser utgår de ofta från sina egna problemlösningsförmågor för att ta sig an uppgifterna. Det ligger i linje med vad Wing (2006) uttrycker angående CT. Nämligen att CT inte bara är förmågor som finns eller inte finns, utan det är rekursivt att använda dessa förmågor. Det vill säga att använda förmågorna kan hjälpa eleverna att utveckla fler förmågor.

Däremot vill jag inte påstå att CT är lösningen på allt, att programmeringsundervisning skulle ändra sättet för alla människor att tänka (jfr Buitrago Flórez et al., 2017). CT har flera olika definitioner (Zhang & Nouris, 2019) och man kan tänka att elevernas tillvägagångssätt i huvudsak kan kopplas till det som brukar benämnas tinkering (Kalelioğlu et al., 2017). Det vill säga att eleverna prövar sig fram på ett lekfullt sätt när de ska programmerar och att det gäller alla tillvägagångssätten. Det som denna studie visar är att andra förmågor som beskrivs och har andra likheter med CT, kan på ett tydligare sätt ge en riktning för varför eleverna ska lära sig programmering men också för lärare, varför de ska undervisa i programmering (Saari et al., 2016).

Genom att se elevernas mer praktiska arbete som ett tillfälle att stödja förmågor som är användbara på fler sätt i ett digitalt samhälle, blir programmeringskunskaperna inte ett mål i sig, utan verktyg för att utveckla kompetenser som är viktiga i ett digitalt samhälle. Något som ligger väl i linje med tankarna om att digital kompetens är nödvändig för att förvärva annan kunskap (Godhe, 2019). I paritet med Turkle och Paperts (1990) forskning från 1990-talet identifierades både abstrakta och konkreta tillvägagångssätt bland eleverna som programmerade på SC. Det visar på att exempelvis programmering med möjlighet till att relativt lätt skapa animeringar och visualiseringar kan ge inträde till nya områden för elever vars främsta sätt att relatera till världen är genom rörelse eller visuellt intryck. Att ge elever denna möjlighet att välja hur de ska lösa sina uppgifter kan även vara en väg för ökat engagemang (Kress, 2015).

Genom analys av hur elevers engagemang kan ta sig i uttryck på olika sätt får vi en förståelse för de specifika situationerna. För att ha möjlighet att engagera sig i en lärsituation är det viktigt att varje elev känner att de har tillgång till kunskapsinnehållet (Cochran et al., 2017). Studien visar att det gäller både yngre och äldre elever. Huruvida eleverna upplever att aktiviteterna under lektionerna som erbjuds är känslomässigt engagerande kan ha att göra med om de uppfattar att uppgifterna är meningsfulla och kognitivt engagerade. Flera elever i studien fick uppmärksamhet just genom sitt positiva känslomässiga engagemang vilket även påverkade deras beteendemässiga engagemang. Att få uppmärksamhet av lärare och kamrater kan leda till att det är lättare att dela med sig av sina tankar och idéer (Harlow & Leak, 2014). Just kommunikation och samarbete är viktiga

komponenter för problemlösning inom programmering (Lewis, 2011; Tsan et al. 2018).

Vidare visar analysen att endast en liten del av lektionerna på SC ägnas åt den andra transformationscykeln i LDS-modellen (Selander, 2017). Merparten av besöket tillbringas med att praktiskt hantera den digitala utrustningen. Detta kan delvis bero på den traditionellt informella inlärmingsmiljön som kännetecknar SC:s (Sasson, 2014). Eftersom en väl fungerade lärproucess är en fråga om att kunna kommunicera (Kress & Selander, 2012) skulle det vara en möjlighet att ge mervärde åt lektionerna genom att ge eleverna mer tid för reflektion och utbyte av erfarenheter. Genom att diskutera vad de olika tillvägagångssätten tränar för typ av förmågor och hur man kan kombinera olika tillvägagångssätt för att lösa problem av varierande karaktär hade eleverna exempelvis kunnat få insikt i fler sätt att arbeta med programmering. Samtidigt är det viktigt att fundera på om målet med ett samarbete mellan SC och skolan i första hand ska vara att eleverna ska erhålla djupa kunskaper i programmering. Kanske kan målet att väcka nyfikenhet och kognitiv- och känslomässigt engagemang (och få elever att känna sig magiska) vara nog. Lärandet kan då följas upp i skolan vid senare tidpunkt.

7.2 Design för lärande i studien

Inför arbetet med denna studie var valet av teoretiskt ramverk ingen självklarhet. Med en bakgrund som pedagog på SC hade jag filosofen och pedagogen John Dewey (Dewey, 2008) i ryggen med en känsla av att "learning by doing" gör jobbet. Ju längre arbete pågick fanns dock ett allt större behov av att förankra mina analyser i en teori. Det var som ett praktiskt och filosofiskt prövande för att förankra ontologi och epistemologi i något som fångade in både praktik och teori. Den designorienterade teori har möjlighet att göra just det. Design kan användas för att förstå lärande, att det handlar om både processer och vad som erkänns som kunskap (Selander, 2017). Det designorienterade tankesättet är också ett verktyg för att utveckla lärmiljöer och arbetsformer (Forsling, 2017; Kress & Selander, 2012; Selander, 2017). Genom *Design för lärande* fick jag således ett redskap som gav möjlighet att grunda arbetet i en teori med en struktur som stödjer ett förändringsriktat arbetssätt. Samtidigt gav LDS-modellen (Selander, 2017) ett analysverktyg för att studera elevers läresekvenser. Även om det inte går att se exakt hur

eller när lärande sker kan man på detta sätt se spåren av lärandet i en individs lärprocess.

Jag har flera gånger, både i handledning och presentation av mitt arbete vid olika konferenser, diskuterat skillnaden mellan det som Selander och Kress (2017) beskriver som design *för* lärande respektive design *i* lärande. Det vill säga de två första aspekterna av didaktisk design (Selander & Kress, 2017). På det sätt jag använder begreppen i denna studie är distinktionen mellan *för* och *i* knutet till vad som är förberett och iscensatt för lärande respektive vad som händer under tiden designen används. På SC är det designat *för* lärande i programmering i form av utrustning, personal och planerade lektioner. Samtidigt så designar även eleverna *för* lärande. Eleverna har olika erfarenheter av programmering. Några har prövat hemma, för andra är det helt nytt. Någon kan ha en bestämd uppfattning om vad programmering innebär, andra har öppet sinne. Det innebär att erfarenheten påverkar förutsättningarna för lärandet i den aktuella situationen, det vill säga designen för lärande under de observerade lektionerna.

Hur eleverna designar *i* lärandet är däremot knutet till vilka tillvägagångssätt de använder när de programmerar och hur de engagerar sig i den aktuella lärsituationen. Hur de designar *i* är en konsekvens bland annat av hur det är designat *för*. Att använda båda begreppen gör att det går att analysera och dra slutsatser om varje del för sig. Design som begrepp inom lärande har alltså flera olika nivåer (Selander, 2017). Samtidigt är det tydligt att design *för* och design *i* hör ihop, då lärande alltid sker i en form av social kontext. Jag menar därför att eleverna inte bara är med och designar, de är *designers* då de designar både *för* och *i* sitt lärande.

Den tredje aspekten av didaktiska design och design för lärande som teori, är gränsdragningen för *erkännandekultur* (Selander & Kress, 2017). Alltså vad som erkänns som kunskap i ett visst sammanhang. Det som styr erkännandekulturen kan alltså exempelvis i skolan vara betygskriterier, i andra lärsituationer är det andra saker som styr. Då formell bedömning av kunskaper inte ingår i SC:s uppdrag kan det uppfattas som att "allting är möjligt" och "ingenting är fel". Därför skulle man kunna ifrågasätta vilket kunskapsinnehåll som eleverna har möjlighet att ta till sig när skolan använder SC. Det vill säga om elevernas lärande i matematik respektive teknik har möjlighet att utvecklas, vilken typ av programmering som används och så vidare. Samtidigt har svenska SC traditionellt ett starkt

band till skolan och följer i de lektioner som erbjuds, intentionerna i grundskolans läroplan (Svenska Science Centers, 2017). När det gäller denna studie var dock programmering vid tidpunkten för datainsamlingen ännu inte ett obligatoriskt moment för grundskolan. Det kan därför finnas en viss diskrepans mellan vad SC *då* erbjöd för undervisning och hade som erkännandekultur gällande programmeringskunskaper för barn och unga och vad som *nu*, genom skrivningarna i gällande läroplan är gällande (Skolverket, 2019a).

7.3 Studiens bidrag

Resultaten i denna studie bidrar med kunskap om hur elever har möjlighet att bli designers för och i sitt lärande i programmering. Studien genomfördes i brytningstiden strax innan det blev obligatoriskt med programmering i hela grundskolan. Då planerade man efter att alla elever var nybörjare oavsett vilken årskurs de gick i (även om inte detta stämde i praktiken, många elever hade provat programmering tidigare). Idag vet man att elever i årskurs ett ofta har provat på någon form av programmeringsövning redan i förskolan. Elever på högstadiet har troligtvis mött programmering på olika nivåer under några år. Dessutom går utvecklingen av digital teknik snabbt. De programmeringsverktyg som finns till hands idag är troligtvis inte de samma om några år. Det primära bidraget med denna studie alltså inte förknippat med vilken typ av programmering som eleverna arbetar med eller hur mycket de lär sig. Denna studie är istället relevant just därför att den inte fokuserar på några speciella programmeringsverktyg utan beskriver hur elever är designers i sin läraprocess kan dessa strukturer gälla för lärande i många sammanhang.

Studien utgör en operationalisering av LDS-modellen när elevers läraprocesser i programmering undersöks. Studien bidrar med kunskap om hur grundskoleelever på nybörjarnivå gör när de ska lära sig programmera. Resultatet från studien visar även hur elever engagerar sig på olika sätt och att de använder förmågor som är viktiga för programmering genom de tillvägagångssätt de använder. Detta i en tid när digital kompetens ses som allt viktigare i dagens samhälle.

Vidare bidrar studien med tankar om hur både SC och grundskolan kan fortsätta att utveckla undervisningen i programmering genom design för lärande. Då området fortfarande är nytt i svensk grundskola finns ett behov

av fortsatt forskning, särskilt då den digitala utvecklingen går i snabb takt och den pedagogiska utvecklingen måste följa med.

7.4 Vidare forskning

En skola för alla, innebär att undervisningen i grundskolan ska vara anpassad efter vars och ens förutsättningar. Då de observerade lektionerna i denna studie var en del av elevernas obligatoriska skoldag kan frågan om alla elevers möjlighet att delta på sina villkor när undervisningen flyttas från skolans lokaler bli viktig att veta mer om. Exempelvis kan instruktioner och samtal (om programmering) utesluta elever då de inte fullt ut förstår språket, inte har samma förförståelse eller av andra orsaker har svårt att delta fullt ut. Därför förslås fortsatt forskning om hur elever upplever sina möjligheter till delaktighet och lärande efter sina förutsättningar när undervisningen flyttas från skolan till SC.

Lärande är en process som pågår på olika sätt och i olika sammanhang. Frågan om *vad* eleverna lär sig har inte behandlats i denna studie. Fortsatt forskning skulle därför kunna innebära att än mer fokusera på kunskapsinnehållet när elever använder olika typer av programvaror och digital utrustning. Det skulle därför även vara intressant att följa om och hur elevers lärande påverkas av den fortsatta digitala utvecklingen och de digitala kompetenser som kommer att krävas i framtiden.

I vidare forskning skulle fortsatt fokus på elevers design *för* och *i* lärande kunna vidga och fördjupa förståelsen för elevers lärprocesser i relation till digital kompetens. Med utgångspunkt i denna studies resultat finns en potential i att utveckla kunskapen och förståelsen för hur elever är designers i sin egen lärprocess.

REFERENSER

- Abrahamsson, C., Malmberg, C., & Pendrill, A-M. (2019). En Delfistudie om lärares uppfattning av elevengagemang i NO-undervisningen. *NorDiNa*, 15(2), 128–144. <https://doi.org/10.5617/nordina.5614>
- Achiam, M., & Sølberg, J. (2016). Nine meta-functions for science museums and science centres. *Museum management and curatorship*, 32(2), 123–143. <http://dx.doi.org/10.1080/09647775.2016.1266282>
- Andersson, S., Heintz, F., Moodeus, G., Vesterlund, J., & Stenliden, L. (2017). Trippel Helix Rapport 3. <http://swedsoft.se/aktiviteter-projekt/strategiska-projekt/trippel-helix-nationell-samling-skolans-digitalisering/> Hämtad 20170308
- Asad, K., Tibi, M., & Raiyn, J. (2016). Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World Journal of Education* 6(5). <http://dx.doi.org/10.5430/wje.v6n5p20>
- Bergdahl, N., Knutsson, O., & Fors, U. (2018). Designing for Engagement in TEL – a Teacher-Researcher Collaboration. *Designs for Learning*, 10(1), 100–111. <http://doi.org/10.16993/dfl.113>
- Bjørndal, C. R. P. (2005). *Det värderande ögat: observation, utvärdering och utveckling i undervisning och handledning*. Liber
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology* 3(2), 77–101. <http://dx.doi.org/10.1191/1478088706qp0630a>
- Brising, E. R. (2019). Programmerar mot nya mål. *Specialpedagogik*, 29 januari <https://specialpedagogik.se/programmerar-mot-nya-mal/> Hämtad 20190201
- Bryman, A. (2016). *Social Research Methods, Fifth Edition*. Oxford University Press.
- Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research*, 87(4), 834–860. <http://dx.doi.org/10.3102/0034654317710096>
- Casey, J. K., Gill, P., Pennington, L., & Mireles, S. V. (2018). Lines, roamers, and squares: Oh my! Using floor robots to enhance Hispanic students' understanding of programming. *Education Information Technology*, 23, 1531–1546. <https://doi.org/10.1007/s10639-017-9677-z>

- Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *Journal of Mathematical Behavior*, 47, 70–80.
<https://doi.org/10.1016/j.jmathb.2017.06.004>
- CETIS (2017). Dokumentation från konferensen Tekniken i skolan i Norrköping 16-17 oktober. <https://liu.se/cetis/konferenser/riks-teknik-2017.shtml> Hämtad 20200417
- Chang, C-K., Yang, Y-A., & Tsai, Y-T. (2017). Exploring the engagement effects of visual programming language for data structure courses. *Education for Information*, 33, 187–200.
<https://doi.org/10.3233/EFI-170108>
- Cochran, K. F., Reinsvold, L. A., & Hess, C. A. (2017). Giving Students the Power to Engage with Learning. *Research in Science Education*, 47, 1379–1401. <https://doi.org/10.1007/s11165-016-9555-5>
- Datainspektionen. (2019). Dataskyddsförordningens grundläggande principer. <https://www.datainspektionen.se/lagar--regler/dataskyddsförordningen/grundlaggande-principer/#Integritet> Hämtad 20190220
- Davidsson, Eva. (2008). Different images of science - a study of how science is constituted in exhibitions. [Doktorsavhandling, Malmö högskola].
- Dawson, E. (2014). Equity in informal science education: developing an access and equity framework for science museums and science centres. *Studies in Science Education*, 50(2), 209–247.
<https://doi.org/10.1080/03057267.2014.957558>
- Department for Education (2013). National curriculum in England: computing programmes of study. Publicerat den 13 september 2013.
- Dewey, J. (2008). *Individ, skola och samhälle*. Natur och Kultur.
- DeWitt, J., & Osborne, J. (2007). Supporting Teachers on Science-focused School Trips: Towards an integrated framework of theory and practice. *International Journal of Science Education*, 29(6), 685–710. <https://doi.org/10.1080/09500690600802254>
- Downey, A. (2015). *Think Python How to think like a computer scientist*. 2nd Edition, Version 2.4.0. Green Tea Press.
- Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, 116, 191–202.
<https://doi.org/10.1016/j.compedu.2017.09.004>
- Falcão, T. P., Barbosa, R. S., & Gomes, T. S. (2017). An Analysis of Interaction Design in Children's Games Based on Computational Thinking. *International Journal on Computational Thinking*, 1(1).
<http://doi.org/10.14210/ijctthink.v1.n1.p16>
- Falk, J. H., & Needham, M. D. (2011). Measuring the Impact of a Science Center on Its Community. *Journal of research in Science Teaching*, 48(1), 1–12.

-
- Fleischer, H., & Kvarnsell, H. (2017). *Digitaliseringen som lyfter skolan. Teori möter praktik*. Gothia Fortbildning.
- Flewitt, R., Hampel, R., Hauk, M., & Lancaster, L. (2017). I: C. Jewitt. *The Routledge Handbook of Multimodal Analysis*. Second edition.
- Forsling, Karin. (2017). *Att överbrygga klyftor i ett digitalt lärandelandskap: design och iscensättning av skriv- och läslärande i förskoleklass och lågstadium*. [Doktorsavhandling, Åbo Akademi].
- Forsling, K. (2019). Designs for Learning: Focus on Special Needs. *Designs for Learning*, 11(1), 108–117.
<http://doi.org/10.16993/dfl.106>
- Fransson, G., Lindberg, J. O., & Olofsson, A. D. (2018). Adequate digital competence – a close reading of the new national strategy for digitalization of the schools in Sweden. *Seminar.net*, 14(2), 217–228.
<https://journals.hioa.no/index.php/seminar/article/view/2982>
Hämtad 20200727
- Fredricks, J. A., Blumenfeld, P. C., & Paris, A. (2004). School engagement: Potential of the concept: State of the evidence. *Review of Educational Research*, 74, 59–119.
<https://doi.org/10.3102/00346543074001059>
- Fredricks J. A., & McColskey, W. (2012). The Measurement of Student Engagement: A Comparative Analysis of Various Methods and Student Self-report Instruments. I: S. Christenson, A. Reschly, & C. Wylie. (eds). *Handbook of Research on Student Engagement*. Springer, Boston, MA DOI https://doi.org/10.1007/978-1-4614-2018-7_37
- Godhe, A. (2019). Digital Literacies or Digital Competence: Conceptualizations in Nordic Curricula. *Media and Communication*, 7(2), 25–35. <http://dx.doi.org/10.17645/mac.v7i2.1888>
- Gunnarson, M. (2019). Att utforska praktisk kunskap med deltagande observation. I: M. Gunnarson (ed.), *Att utforska praktisk kunskap: Undersökande, prövande och avtäckande metoder*, 225–260. Södertörns högskola.
- Halmstad kommun. (2019). Valhallaskolans testmiljö.
<https://www.halmstad.se/barnutbildning/kvalitetochutveckling/projekt/realclassroomlab/valhallaskolanstestmiljo.24051.html> Hämtad 20190201
- Halverson, L. R., & Graham, C. R. (2019). Learner engagement in blended learning environments: A conceptual framework. *Online Learning*, 23(2), 145–178. <https://doi.org/10.24059/olj.v23i2.1481>
- Harlow, D. B., & Leak, A. E. (2014). Mapping students' ideas to understand learning in a collaborative programming environment. *Computer Science Education*, 24(2–3), 229–247.
<http://dx.doi.org/10.1080/08993408.2014.963360>
- Heintz, F., Vesterlund, J., Stenliden, L., & Andersson, S. (2016). Trippel Helix Rapport 2. <http://swedsoft.se/aktiviteter-projekt/strategiska->

projekt/trippel-helix-nationell-samling-skolans-digitalisering/

Hämtad 20170308

- Heddens, J. (1986). Bridging the gap between the concrete and the abstract. *The Arithmetic Teacher*, 33(6), 14–17.
- Hymers, D., & Newton, G. (2019). Investigating student engagement in first-year biology education: A comparison of major and non-major perception of engagement across different active learning activities. *The Canadian Journal for the Scholarship of Teaching and Learning*, 10(1). <https://doi.org/10.5206/cjsotl-rcacea.2019.1.7993>
- Insulander, E., Kjällander, S., Lindstrand, F., & Åkerfeldt, A. (red). 2017. *Didaktik i omvandlingens tid. Text, representation, design*. Liber.
- Itzek-Greulich, H., & Vollmer, C. (2017). Emotional and Motivational Outcomes of LabWork in the Secondary IntermediateTrack: The Contribution of a Science Center Outreach Lab. *Journal of research in Science Teaching*, 54(1), 3–28. <http://doi.org/10.1002/tea.21334>
- Jahnke, I., & Kumar, S. (2014). Digital Didactical Designs: Teachers' Integration of iPads for Learning-Centred Processes. *Journal of Digital Learning in Teacher Education*, 30(3), 81–88. <http://doi.org/10.1080/21532974.2014.891876>
- Jewitt, C. (2017). *The Routledge Handbook of Multimodal Analysis*. Second edition.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200–210. <http://doi.org/10.1016/j.chb.2015.05.047>
- Kalelioğlu, F., & Gülbahar Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33–50.
- Kalelioğlu, F., Gülbahar, Y. & Doğan, D. (2017). Teaching How to Think Like a Programmer: Emerging Insights. I: H. Özçınar, G. Wong, & H. T. Öztürk (Eds.) *Teaching Computational Thinking in Primary Education*, (18-35). Information Science Reference. <http://doi.org/10.4018/978-1-5225-3200-2.ch002>
- Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4, 583–596.
- Kjällander, S. (2011). *Designs for Learning in an Extended Digital Environment. Case Studies of Social Interaction in the Social Science Classroom*. [Doktorsavhandling, Stockholms universitet].
- Kjällander, S., Åkerfeldt, A., Mannila, L., & Parnes, P. (2018). Makerspaces Across Settings: Didactic Design for Programming in Formal and Informal Teacher Education in the Nordic Countries, *Journal of Digital Learning in Teacher Education*, 34(1), 18-30. <http://doi.org/10.1080/21532974.2017.1387831>
- Kjällander, S., Åkerfeldt, A., & Petersen, P. (2016). Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola.

-
- <https://www.divaportal.org/smash/get/diva2:1426299/FULLTEXT01.pdf> Hämtad 20200618
- Klem, A. M., & Connell, J. P. (2004). Relationships matter: linking teacher support to student engagement and achievement. *Journal of School Health*, 74(7), 262-273. <https://doi.org/10.1111/j.1746-1561.2004.tb08283.x>
- Komtek (u.å.). Skapa kul teknik! <https://komtek.se/> Hämtad 20200408
- Kong, S.-C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude and programming empowerment in computational thinking education. *Computers & Education*, 127, 178-189. <https://doi.org/10.1016/j.compedu.2018.08.026>
- Kress, G. (2000). "You've Just Got to Learn How to See": Curriculum Subjects, Young People and Schooled Engagement with the World. *Linguistics and Education*, 11, 401-415. [http://doi.org/10.1016/S0898-5898\(00\)00030-9](http://doi.org/10.1016/S0898-5898(00)00030-9)
- Kress, G. (2015). Semiotic work: Applied Linguistics and a social semiotic account of Multimodality. *AILA Review*, 28. <https://doi.org/10.1075/aila.28.03kre>
- Kress, G., & Selander, S. (2012). Multimodal design, learning and cultures of recognition. *Internet and Higher Education* 15, 265-268. <http://doi.org/10.1016/j.iheduc.2011.12.003>
- Kubacki, G., Elsholz, M., & Trefzger, T. (2012). Science Center im Spannungsfeld zwischen Wissenvermittlung und Freizeitspaß. *Didaktik der Physik*. Frühjahrslagung-Mainz.
- Larsson, S. (2005). Om kvalitet i kvalitativa studier. *Nordisk Pedagogik*, 25(1), 16-35.
- Leijon, M. (2010). Att spåra tecken på lärande: *Mediereception som pedagogisk form och multimodalt meningsskapande över tid*. [Doktorsavhandling, Malmö högskola]. Malmö.
- Leijon, M., & Lindstrand, F. (2012). Socialsemiotik och design för lärande: två multimodala teorier om lärande, representation och teckenskapande. *Pedagogisk Forskning i Sverige*, 17(3-4).
- Levinsen, K. T., & Sørensen, B. H. (2019). Teachers' Designs for Learning Practices when Designing for Students as Learning Designers. *Designs for Learning*, 11(1), 30-39. <https://doi.org/10.16993/dfl.111>
- Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education*, 21(2), 105-134.
- Loseke, D. R. (2017). *Methodological Thinking. Basic principles of social research design*. Second edition. SAGE Publications, Inc.
- Lykke, M., Coto, M., Jantzen, C., Mora, S., & Vandel, N. (2015). Motivating Students Through Positive Learning Experiences: A Comparison of Three Learning Designs for Computer Programming Courses. *Journal of Problem Based Learning in Higher*

- education, 3(2), 80–108.
<http://dx.doi.org/10.5278/ojs.jpblhe.voio.1130>
- Makers of Sweden. (u.å.). Vad är ett makerspace?
<http://www.makersofsweden.se/> Hämtad 20190201
- Mannila, L. (2017). Att undervisa i programmering i skolan. Varför, vad och hur? Studentlitteratur.
- Martin, A. J., Durksen, T. L., Williamson, D., Kiss, J., & Ginns, P. (2016). The Role of a Museum-Based Science Education Program in Promoting Content Knowledge and Science Motivation. *Journal of Research in Science Teaching*, 53(9), 1364–1384.
<http://dx.doi.org/10.1002/tea.21332>
- Melander Bowden, H. (2019). Problem-solving in collaborative game design practices: epistemic stance, affect, and engagement. *Learning, Media and Technology*, 44(2), 124–143.
<https://doi.org/10.1080/17439884.2018.1563106>
- Mladenovic, M., Boljat, I., & Žanko, Ž. (2018). Comparing Loops Misconceptions in Block-Based and Text-Based Programming Languages at the K-12 Level. *Education and Information Technologies*, 23(4), 1483–1500. <https://doi.org/10.1007/s10639-017-9673-3>
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020) Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17. <https://doi.org/10.1080/20004508.2019.1627844>
- Ouahbi, I., Kaddaria, F., Darhmaouib, H., Elachqara, A., & Lahminea, S. (2015). Learning basic programming concepts by creating games with Scratch programming environment. *Procedia - Social and Behavioral Sciences*, 191, 1479–1482.
<http://doi.org/10.1016/j.sbspro.2015.04.224>
- Pareto, L., & Willermark, S. (2019). TPACK In Situ: A Design-Based Approach Supporting Professional Development in Practice. *Journal of Educational Computing Research*, 57(5), 1186–1226.
<https://doi.org/10.1177/0735633118783180>
- Pedretti, E. G. (2004). Perspective on learning through research on critical issues-based Science Center exhibitions. *Wiley InterScience*.
- Piqueras, J., Hamza, K. M., & Edvall, S (2008) The Practical Epistemologies in the Museum. *Journal of Museum Education*, 33(2), 153–164. <https://doi.org/10.1080/10598650.2008.11510596>
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365–376.
<https://doi.org/10.1016/j.compedu.2018.10.005>
- Portelance, D. J., Strawhacker, A. M., & Bers, M. U. (2016). Constructing the Scratch Jr programming language in the early childhood classroom. *International Journal of Technology and Design Education*, 26, 489–504. <http://doi.org/10.1007/s10798-015-9325-0>

-
- Regeringskansliet. (2015). Gör Sverige i framtiden – digital kompetens. Delbetänkande av Digitaliseringskommissionen. SOU 2015:28 Stockholm
- Rolandsson, L. (2015). *Programmed or not. A study about programming teachers' beliefs and intentions in relation to the curriculum*. [Doktorsavhandling, KTH].
- Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68–78.
- Sáez-López, J-M., Román-González M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129–141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Saari, E. M., Blanchfield, P., & Hopkins, G. (2016). Computational Thinking -A Tool to Motivate Understanding in Elementary School Teachers. Conference paper. https://doi.org/10.1007/978-3-319-29585-5_20
- Samuelsson, U. (2014). *Digital (o)jämlighet. IKT-användning i skolan och elevers tekniska kapital*. [Doktorsavhandling, Högskolan för lärarutbildning och kommunikation].
- Sasson, I. (2014). The role of informal science centre in science education: Attitudes, skills and self-efficacy. *Journal of Technology and Science Education*, 4(3), 167–180. <https://dx.doi.org/10.3926/jotse.123>
- Shaby, N., Ben-Zvi Assaraf, O., & Tal, T. (2017). The Particular Aspects of Science Museum Exhibits That Encourage Students' Engagement. *J Sci Educ Technol*, 26, 253–268 <https://doi.org/10.1007/s10956-016-9676-7>
- Selander, S. (2007). Didaktisk design - den dubbla utmaningen. *Digital Kompetanse*, 2(3), 162–172.
- Selander, S. (2008). Designs for Learning - A Theoretical perspective. *Designs for Learning*, 1(1), 4–22. <http://doi.org/10.16993/dfl.5>
- Selander, S. (2017). *Didaktiken efter Vygotskij. Design för lärande*. Liber.
- Selander, S., & Kress, G. (2017). *Design för lärande- ett multimodalt perspektiv*. Andra upplagan. Studentlitteratur.
- Selander, S., & Rostvall, A-L. (2010). Design och meningsskapande - en inledning. I: A-L. Rostvall, & S. Selander (eds.). *Design för lärande*. Norstedts.
- SFS 2010:800. Skollag. https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/skollag-2010800_sfs-2010-800
- Siddiq, F. (2018). A Comparison between digital competence in two Nordic countries' national curricula and an international framework: Inspecting their readiness for 21st century education. *Seminar.net*,

- 14(2), 144-159.
<https://journals.hioa.no/index.php/seminar/article/view/2977>
 Hämtad 20200413
- Skansholm, J. (2018). *Programmera på riktigt. För lärare på högstadiet*. Studentlitteratur.
- Skolinspektionen. (2019). Digitala verktyg i undervisningen - matematik och teknik i årskurs 7-9. Granskningsrapport Dnr 400-2018-6938
- Skolverket. (u.å). Lärportalen. Hämtad 20200826
<https://larportalen.skolverket.se/#/>
- Skolverket. (2016). Digital kompetens och programmering ska stärkas i skolan. Hämtad 20190219 <https://www.skolverket.se/laroplaner-amnen-och-kurser/nyhetsarkiv/nyheter-2016/nyheter-2016-1.247899/digital-kompetens-och-programmering-ska-starkas-i-skolan-1.247906>
- Skolverket. (2017a). Presentationer från workshop i teknik- ht 2017. Hämtad 20190304
<https://www.skolverket.se/temasidor/digitalisering/konferenser-om-skolans-digitalisering#h-Programmeringimatematikochteknikgrundskolan>
- Skolverket. (2017b). Få syn på digitaliseringen på grundskolenivå–Ett kommentarmaterial till läroplanerna för förskoleklass, fritidshem och grundskoleutbildning. Hämtad 20200413
<https://www.skolverket.se/publikationer?id=3783>
- Skolverket. (2019a). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2011: Reviderad 2019 (6 uppl.)*.
<https://www.skolverket.se/getFile?file=4206>
- Skolverket. (2019b). Statsbidrag för Science Center 2019.
<https://www.skolverket.se/skolutveckling/statsbidrag/statsbidrag-for-science-center-2019> Hämtad 20190219
- Streicher, Barbara (2016). Science Center und ihre Aktivitäten als Schnittstelle zwischen Universität und Erwachsenenbildung. *Magazin erwachsenenbildung at Das Fachmedium für Forschung, Praxis und Diskurs*.
<http://www.erwachsenenbildung.at/magazin/16-27/meb16-27.pdf>.
 Hämtad 181123
- Svenska Science Centers. (2017). Stadgar.
<http://media.fssc.se/2017/03/Stadgar-2017.pdf> Hämtad 20190219
- Svenska Science Centers. (2018). Presentation av Science Centers 2018.
<https://fssc.se/wp-content/uploads/2018/04/Presentation-av-Science-Centers-2018-.pdf> Hämtad 20190219
- Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of educational computing*, 57(1), 3-31. <https://doi.org/10.1177/0735633117743918>

-
- Tsan, J., Rodríguez, F. J., Boyer, K. E., & Lynch, C. (2018). "I Think We Should. . .": Analyzing Elementary Students' Collaborative Processes for Giving and Taking Suggestions. In Proceedings of SIGCSE '18: The 49th ACM Technical Symposium on Computing Science Education, Baltimore, MD, USA, February 21–24, 2018 (SIGCSE '18), 6 pages. <https://doi.org/10.1145/3159450.3159507>
- Turkle, S., & Papert, S. (1990). Epistemological Pluralism: Styles and Voices within the Computer Culture. *Signs*, 16(1), 128–157.
- Twining, P., Heller, R. S., Nussbaum, M., & Tsai, C-C. (2017). Some guidance on conducting and reporting qualitative studies. *Computers & Education*, 106, A1-A9. <https://doi.org/10.1016/j.compedu.2016.12.002>.
- Vetenskapsrådet. (2017). *God forskningssed*. Stockholm: Vetenskapsrådet
- Willermark, S. (2018). *Digital Didaktisk Design. Att utveckla undervisningspraktiken i och för en digitaliserad skola*. [Doktorsavhandling, Högskolan Väst].
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. <http://doi.org/10.1145/1118178.1118215>
- Winter, G. (2000). A Comparative Discussion of the Notion of 'Validity' in Qualitative and Quantitative Research. *The Qualitative Report*, 4(3), 1-14. <https://nsuworks.nova.edu/tqr/vol4/iss3/4>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>
- Åkerfeldt, A. (2014). *Didaktisk design med digitala resurser: En studie av kunskapsrepresentationer i en digitaliserad skola*. [Doktorsavhandling, Stockholms universitet].
- Åkerfeldt, A., Kjällander, S., & Selander, S. (2018). *Programmering introduktion till digital kompetens i grundskolan*. Liber.

APPENDIX

Appendix 1

Programmeringsverktyg som används på SC under de observerade lektionerna.

Alla länkar kontrollerade 1 september 2020

Arduino: <https://www.arduino.cc/>

KOJO: <http://www.lth.se/programmera/bakgrund/>
<http://www.lth.se/programmera/installera/>

LEGO® NXT and LEGO® ew3: <https://www.lego.com/sv-se/mindstorms/learn-to-program>

Makey makey: <http://makeymakey.com/>

Scratch: <https://scratch.mit.edu/>

Scratch JR: <https://www.scratchjr.org/>

Appendix 2

Brev till barn och vårdnadshavare

Våren 2017

Hej

Jag heter Maria Sparf och arbetar sedan 12 år tillbaka på XXX, YYYs Science center. Vid sidan av arbetet är jag en av Jönköpings kommuns doktorander inom pedagogik vid Linköpings universitet.

Under våren 2017 påbörjar jag mitt arbete med att samla material till min forskning. Den kommer i huvudsak handla om hur ett Science Center kan stötta skolors undervisning inom naturvetenskap och teknik. Läs mer om Science center på <http://fssc.se/> När klassen besöker.... kommer jag att finnas med för att observera elever ur klassen när de arbetar i grupp eller i helklass. Jag kommer även att prata med elever kring hur de upplever besöket. Delar av lektionen kommer att filmas.

Det insamlade materialet kommer att användas av mig och mina forskarkollegor för att ligga till grund för forskning inom pedagogik och behandlas helt anonymt, inga enskilda elever kommer att identifieras med vare sig namn eller skola.

Deltagandet är frivilligt och kan avbrytas under tiden det pågår. All data (insamlat material) kommer att sparas lösenordskyddat.

För att genomföra detta behöver jag ert och barnets godkännande att vara med vid studien.

Vill du veta mer innan du bestämmer dig så kontakta mig gärna:

fornamn.efternamn@stad.se

Med vänliga hälsningar Maria

Jag kan tänka mig att vara med och bli filmad när jag arbetar:

Elevens underskrift:

Jag godkänner att mitt barn deltar i undersökningen:

Vårdnadshavares underskrift:

Vårdnadshavares underskrift:

Appendix 3

Brev till lärare

Våren 2017

Hej

Jag heter Maria Sparf och arbetar sedan 12 år tillbaka på XXX, YYYs Science center. Vid sidan av arbetet är jag en av Jönköpings kommuns doktorander inom pedagogik vid Linköpings universitet.

Under våren 2017 påbörjar jag mitt arbete med att samla material till min forskning. Den kommer i huvudsak handla om hur ett Science Center kan stötta skolors undervisning inom naturvetenskap och teknik. Läs mer om Science center på <http://fssc.se/> Jag planerar att under dagen den 8 juni finnas med för att observera elever när de arbetar i grupp eller i helklass med programmering på Jag kommer även att prata med elever kring hur de upplever besöket. Delar av lektionen kommer att filmas.

Det insamlade materialet kommer att användas av mig och mina forskarkollegor för att ligga till grund för forskning inom pedagogik och behandlas helt anonymt, inga enskilda elever eller lärare kommer att identifieras med vare sig namn eller skola. Deltagandet är frivilligt och kan avbrytas under tiden det pågår. All data kommer att sparas lösenordskyddat.

Jag vill också veta mer om vad du som lärare tänker om att utnyttja ... i ditt arbete. Vilka för- och eventuella nackdelar ser du med detta? Jag skulle därför gärna intervjua dig en stund efter klassens genomförda lektion. Om det känns bättre så kan jag vid ett tillfälle du bestämmer, ringa upp dig. Även dessa uppgifter kommer att behandlas helt anonymt och du kan välja att avbryta deltagandet under pågående arbete.

Vill du veta mer innan du bestämmer dig så kontakta mig gärna:
förnamn.efternamn@stad.se 036-000000

Med vänliga hälsningar Maria

Jag ställer gärna upp på en intervju i samband med att min klass besöker ... ☐

Jag vill att du ringer upp mig för en intervju efter klassens besök på ... ☐

Kontaktuppgifter:

Namn:

Mejladress:

Telefon:

TIDIGARE UTGIVNA RAPPORTER

- 243 MATWEJEFF, SUSANNA. Svenskfödda adopterades sökprocess. Januari 2005. (Licentiatavhandling)
- 244 SAMUELSSON, JOAKIM. Lärarstudenters erfarenheter av matematikundervisning. Vad händer med elever när de inte förstår? Maj 2005.
- 245 ÖFVERSTRÖM CHRISTEL. Upplevelse, inlevelse och reflektion - drama som en aktiv metod i lärandet. September 2006. (Licentiatavhandling) ISBN: 91-85643-71-8
- 246 ERIKSSON-GUSTAVSSON, ANNA-LENA, SAMUELSSON, JOAKIM. Didaktiska samtal i specialpedagogiska kontexter. En studie av undervisning i grundläggande svenska och matematik. Februari 2007. ISBN: 978-91-85715-68-8
- 247 ÖSTERSTRÖM, STEFAN. Om konsten att överbrygga gränser. En fallstudie om kommunal äldreomsorg och samspelet med andra organisationer. Maj 2007. ISBN: 978-91-85895-04-5.
- 248 ENGSTRÖM, ARNE, MAGNE, OLOF. Medelsta-matematik IV. En empirisk analys av Skolverkets förslag till mål att uppnå i matematik för årskurs 3. Mars 2008. ISBN: 978-91-7393-918-8.
- 249 THORNBERG, ROBERT. Vilka värden elever enligt lärare ska få med sig från skolan. April 2008. ISBN: 978-91-7393-909-6.
- 250 COLNERUD, GUNNEL, KARLSSON, INGRID, SZKLARSKI, ANDRZEJ. Alltid redo. Lärarstudenters handlingsberedskap för varierande uppgifter i klassrummet. 2008. ISBN: 978-91-7393-784-9.
- 251 SZCZPANSKI, ANDERS. Handlingsburen kunskap Lärares uppfattningar om landskapet som lärandemiljö. 2008. (Licentiatavhandling) ISBN: 978-91-7393-889-1.
- 252 THORNBERG, ROBERT. Ett resursteams samverkan med skola, elever och föräldrar: förtjänster, hinder och utmaningar. 2009. ISBN 978-91-7393-599-9 (Finns bara i elektronisk version).
- 253 HAMMAR CHIRIAC, EVA. Släpp tankarna loss – det är nytt. Kvalitets-granskning av ett reformarbete. Ny speciallärarutbildning. 2009. ISBN: 978-91-7393-541-8
- 254 ERIKSSON-GUSTAVSSON, ANNA-LENA, HOLME, LOTTA. Att göra olika lika: Universitetslärares uppfattningar om och erfarenheter av undervisning av funktionshindrade studenter. 2009. ISBN: 978-91-7393-529-6
- 255 LÖNEBRINK, THOMAS. Processer i växelverkan: En grundad teoretisk modell om landstingskommunal samverkan för personer med psykiska funktionshinder. 2010 (Licentiatavhandling). ISBN: 978-91-7393-242-4

- 256 HARLIN, EVA-MARIE. Överraskning och Reflektion. Lärarstudenters lärande från egen videoinspelad undervisning. 2010 (Licentiatavhandling). ISBN: 978-91-7393-228-8
- 257 ERIKSSON-GUSTAVSSON, ANNA-LENA. ”Det är tufft att plugga... men jag tror att jag klarar det”: En studie om akademiska studier och skriftspråkliga svårigheter. 2011. ISBN: 978-91-7393-183-0
- 258 SAMUELSSON, JOAKIM. Den skicklige matematikläraren. 2013. ISBN: 978-91-7519-609-1
- 259 FREDRIKSSON MÅRTENSSON, ÅSA. Handledare och handledning: gymnasial yrkesutbildning på förskola. 2014 (Licentiatavhandling). ISBN: 978-91-7519-269-7
- 260 WALLIN, JESSICA. Entreprenörskap i skolan: Formulering och transformering av GY11 inom gymnasieskolans hantverksprogram. 2014 (Licentiatavhandling). ISBN: 978-91-7519-263-5
- 261 ALM, F, JUNGERT, T, THORNBERG, R: Nyantagna lärarstudenters motiv, motivation, självförtroende och akademiska engagemang. 2014. ISBN: 978-91-7519-384-7 (finns bara i elektronisk version).
- 262 KEREKES, KLARA. Undervisning om växande geometriska mönster: En variationsteoretisk studie om hur lärare behandlar ett matematiskt innehåll på mellanstadiet. 2014. (Licentiatavhandling) ISBN: 978-91-7519-135-5
- 263 BERGSETH, BRITA. Vägledande eller vilseledande?: Kvalitetsmätning och ranking av universitet och högskolor. 2015 (Licentiatavhandling). ISBN: 978-91-7685-937-7.
- 264 SVEIDER, CECILIA. Lärares och elevers användande av laborativt material i bråkundervisningen i skolår 4-6: Vad görs möjligt för eleverna att erfara. 2015 (Licentiatavhandling) ISBN: 978-91-7685-827-1
- 265 AMINOFF, CHRISTINA. Samtals- och skriftspråksorienterade lärarledda aktiviteter i förskoleklass. 2017 (Licentiatavhandling) ISBN: 978-91-7685-556-0
- 266 SUNDSTRÖM, PETHER. Coachande nätbaserade samtal om matematik. En studie om hur matematikcoacher stöttar elever vid problemlösning. 2020 (Licentiatavhandling) ISBN: 978-91-7929-842-5

Institutionen för beteendevetenskap och lärande
LiU-PEK-R-267

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se