



Estetisk-filosofiska fakulteten
Pedagogiskt arbete

Mikael Segolsson

Programmeringens intentionala objekt

Nio elevers uppfattningar av programmering

Mikael Segolsson

Programmeringens intentionala objekt

Nio elevers uppfattningar av programmering

Mikael Segolsson. *Programmeringens intentionala objekt: Nio elevers uppfattningar av programmering*

Licentiatavhandling

Karlstad University Studies 2006:50

ISSN 1403-8099

ISBN 91-7063-081-X

© Författaren

Distribution:

Karlstads universitet

Estetisk-filosofiska fakulteten

Pedagogiskt arbete

651 88 KARLSTAD

SVERIGE

054-700 10 00

www.kau.se

Tryck: Universitetstryckeriet, Karlstad 2006

ABSTARCT

When it comes to the way we work in the postmodern society of today, essential parts of our tasks are connected to computers in one way or another. The Robots in the industry are one example where traditional human work, concretely and physically, has been replaced by programming and supervising processes in the production. These stages of development make new demands on education and recruiting of labors. The recruiting of qualified labors has turned out to be problematic, and even though extensive efforts has been made in order to increase the interest of technological education, the interest is still weak in Sweden.

This study presents the research question: *What do pupils focus their attention on when describing programming?* The study belongs to a survey conducted on how pupils conceptualise when working with programmable construction material during technology lessons. Working with programmable construction material in schools is one example of computer based learning by means of technological components. The aim is to teach students to visualise technology as a phenomenon in society, as well as introducing them to work with computers.

The scientific foundation of the study is grounded in the field of life-world phenomenology with methodology prompting in phenomenography. A fundamental concept in phenomenology, as well as in this study, is intentionality. Another central concept is experience as it is used in phenomenography. The purpose is to describe what pupils, year 8 and 9, focus their attentions on when they are programming – what does the students' intentional object look like when describing a program?

The results of the analyses, one can conclude that four qualitatively separate categories of pupils' intentional object have appeared. The programming is described with focus on: *systematic grouping of the program, to remember another solution of program, the movement of the robot or the solution of the task.*

SAMMANFATTNING

I dagens postmoderna samhälle är väsentliga delar i vårt arbete relaterade till datorer i någon form. Robotiseringen inom industrin utgör ett exempel på hur det mänskliga arbetet förskjutits från det konkret fysiska till att bestå av programmering och övervakning av produktionsprocesser. Denna utveckling ställer nya krav på både utbildning och rekrytering av arbetskraft, något som visat sig vara svårt eftersom intresset för tekniska utbildningar är svagt i Sverige, trots stora satsningar på ökad rekrytering inom de tekniska utbildningarna.

I föreliggande studie presenteras forskningsfrågan: *Vad riktar eleverna sin uppmärksamhet mot i sina beskrivningar av programmeringshandlingen?* Studien är en del av en kartläggning av hur elever begreppsbygger då de arbetar med programmerbart konstruktionsmaterial i teknikundervisningen. Arbetet med programmerbara robotar i skolan är ett exempel på hur tekniska komponenter integreras med datorstött lärande för att ge eleverna en bild av teknik som fenomen i samhället och som introduktion till datorarbete inom skola.

Studiens vetenskapsteoretiska utgångspunkt ligger inom livsvärldsfenomenologi med metodisk ansats i fenomenografi. Ett grundläggande begrepp inom fenomenologi och även för denna studie är intentionalitet. Centralt är också uppfattningsbegreppet så som det används inom fenomenografi. Syftet med uppsatsen är att beskriva vad elever i skolår 8 och 9 riktar sin uppmärksamhet mot när de programmerar – hur elevernas intentionala objekt visar sig när de beskriver sina program.

Analysen har resulterat i fyra beskrivningskategorier av elevernas intentionala objekt. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot: *Systematisk indelning av programmet, att komma ihåg tidigare programlösningar, robotens rörelser och att lösa uppgiften.*

INNEHÅLL

INLEDANDE ORD	7
1. INLEDNING	9
UTGÅNGSPUNKTER I FRÅGESTÄLLNINGEN	13
FORSKNINGSFRÅGAN	15
SYFTE	15
POSITIONERING	16
BESKRIVNING AV KONSTRUKTIONSMATERIALET	17
2. TIDIGARE FORSKNING	21
LÄRANDE MED KONSTRUKTIONSMATERIAL	22
LÄRANDE OCH PROGRAMMERING	27
3. OM PROGRAMMERING	31
PROGRAMMERING – EN ÖVERSIKT	31
GRUNDERNA I PROGRAMMERING	33
PROGRAMMERING – EN FORM AV PROBLEMLÖSNING	36
PROBLEMLÖSNING OCH PROGRAMMERING – NÅGRA KRITISKA ASPEKTER	38
PROGRAMMERING MED GRAFISKA KOMPONENTER	40
DIREKT MANIPULATION	41
LÄRANDE GENOM PROGRAMMERING	42
ÖVERFÖRBARA KUNSKAPER TILL ANDRA OMRÅDEN	43
4. TEORETISKA UTGÅNGSPUNKTER	45
LIVSVÄRLDSFENOMENOLOGI	45
ORIENTERING I FENOMENOLOGIN	46
BEGREPPET INTENTIONALITET	49
MEDERFARENHET	50
DET INTENTIONALA OBJEKTET – INTE ETT TING	51
INTENTIONALITETENS KATEGORISKA OBJEKT	51
LIVSVÄRLDSFENOMENOLOGINS BETYDELSE FÖR STUDIEN	52

KUNSKAP OCH LÄRANDE	53
SAMMANFLÄTADE BEGREPP	54
BEGREPPET HANDLING	55
REFLEKTION I HANDLING	56
5. METOD	59
FENOMENOGRAFI	59
VAL AV METOD	60
TEORETISK FÖRANKRING	61
CENTRALA BEGREPP INOM FENOMENOGRAFIN	62
FRÅN ATT UPPFÄTTA TILL ATT ERFARA	63
KRITISKA ASPEKTER	64
DEN FENOMENOGRAFISKA ARBETSGÅNGEN	65
DEN KVALITATIVA FORSKNINGSINTERVJUN	67
BESKRIVNING AV DATAINSAMLINGEN	68
MÖTET MED ELEVERNA	69
INTERVJUN	70
STIMULATED RECALL SOM STÖD VID DATAINSAMLING	71
URVAL	72
ETISKA ÖVERVÄGANDEN	73
6. RESULTAT	75
KAPITLET'S DISPOSITION	76
RESULTATÖVERSIKT	76
UTFALLSRUMMET	78
A. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT SYSTEMATISK INDELNING AV PROGRAMMET	78
B. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT ATT KOMMA IHÅG TIDIGARE PROGRAMLÖSNINGAR	80
C. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT ROBOTENS RÖRELSER	81
D. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT ATT LÖSA UPPGIFTEN	83

REFLEKTIONER KRING UTFALLSRUMMET	85
FÖRMÅGAN ATT SKAPA INRE BILDER	87
OBSERVATIONER UNDER INTERVJUERNA	88
KUNSKAPSBIDRAG	90
7. DISKUSSION	93
REFLEKTIONER KRING SYFTE OCH METOD	94
REFERENSER	97

BILD- OCH FIGURFÖRTECKNING

BILD 1: Roboten som eleverna programmerade i studien	18
BILD 2: Delar av en bana där roboten ska utföra olika uppdrag	18
BILD 3: Illustration av Merleau-Pontys klassiska exempel på subjektet och objektets ömsesidiga beroende	48
FIGUR 1: Booths fyra kategorier av kvalitativt skilda sätt som studenter erfar programmering av ett rekursionsproblem. Kategorierna är delar av resultatet	27
FIGUR 2: Exempel på hur grundoperationerna sekvens, selektion och iteration används i studiens programmeringsmiljö	35
FIGUR 3: Schematisk beskrivning av den fenomenografiska arbetsgången	66
FIGUR 4: Schematisk bild över studiens fyra beskrivningskategorier i utfallsrummet ..	77

INLEDANDE ORD

Solen skiner och denna septemberdag visar sig på sitt bästa humör. Jag tänker på de dryga två år som passerat i min forskarutbildning. Jag tänker också på föreliggande text som på ett eller annat sätt representerar att jag kommit halvvägs i utbildningen. Det svåraste är att sätta punkt. En utbildning medför – i de flesta fall – en progressiv utveckling, vilket innebär att när det är dags att knyta ihop resultatet av det gångna arbetet upptäcker man att det finns en del som skulle ha gjorts annorlunda. Och tur är väl det!

Jag vill tacka mina handledare Anders Arnqvist, Tomas Kroksmark, Sten Carlsson och AnnBritt Enochsson för de utvecklande och lärorika synpunkter som jag fått under vägen. Tack till min familj och särskilt min son Jacob som vid fem års ålder gav mig visdomsorden: ”Fel tanke, försök igen”. Det finns en positiv anda i de orden och jag kommer alltid att bära dem med mig. Sist men inte minst vill jag rikta ett varmt tack till de elever och lärare på de skolor där studien har bedrivits. Utan er hjälp hade denna studie inte kunnat genomföras.

Mot nya utmaningar!

Jönköping den 18 september 2006

Mikael Segolsson

1. INLEDNING

En sak som människor har gemensamt är att de alla är olika.

(Marton & Booth, 2000, s.15)

Under senare delen av 1990-talet har den tekniska utvecklingen i hela Västvärlden varit så pass genomgripande att vi talar om en infrastrukturell vändning bort från industrisamhället till vad som i olika sammanhang benämns IT-samhället. Väsentliga delar i arbetets innehåll under industrisamhället har ersatts av arbetsuppgifter som är relaterade till datorer i någon form. Detta ser vi inte minst inom industrin där robotar¹ används i allt större utsträckning och den mänskliga arbetsinsatsen oftast består av att programmera, kontrollera och övervaka. Vi ser även i vår vardag att robotar genom filmer, leksaker och andra produkter får en allt större plats. Vi står utan tvekan på tröskeln till ett robotintegrerat samhälle och denna utveckling ställer nya krav på både utbildning och rekrytering av arbetskraft. En problematik i denna process är det svaga intresset för tekniska utbildningar i Sverige. Trots stora förändringar av prioriteringar av de tekniska ämnenas betydelse vid våra universitet förblir attraktionskraften låg. Ett sätt att möta denna samhällsproblemik är att intressera elever redan i grundskolan för teknik och IT-frågor. I föreliggande studie behandlas en aspekt av lärande, lärande av grafisk programmering², som är knutet till elever som arbetat med programmerbart konstruktionsmaterial under tekniklektionerna i grundskolan.

Grunden till denna studie utgörs av en undran över hur elever begreppsbildar då de programmerar robotar. Med begreppsbildning menar jag den process där eleverna lär sig att se något som centralt, där vissa aspekter framträder tydligare än andra när det gäller lärandet av något. Intresset i denna studie är riktat mot att förstå och beskriva hur eleverna uppfattar programmeringshandlingen och vad eleverna riktar sin uppmärksamhet mot i sina beskrivningar.

¹ Ordet *robot* kommer ursprungligen från det slaviska ”roboťa”, vilket betyder träl eller tvingad arbetskraft (Nordin & Wilde, 2003).

² Med grafisk programmering menar jag en programmeringsmiljö där bilder (ikoner) sätts samman till en sekvens av programinstruktioner. Se vidare kapitel 3 *Om programmering*.

Studien är grundad i en livsvärldsfenomenologisk ansats med stöd i fenomenografiska teorier³ och metodiska tillvägagångssätt för att intervjua, analysera och beskriva insamlad data. Denna kvalitativa studie inryms således inom den fenomenografiska ramen för empiriska studier, med vetenskapsteoretiskt stöd i livsvärldsfenomenologin. Livsvärlden är pluralistisk och innebär att vi erkänner världen som en komplex och oreducerbar verklighet (Bengtsson, 1999). Avsikten är – som nämnts – att förstå och beskriva vad eleverna riktar sin uppmärksamhet mot när de beskriver sina programmeringshandlingar där ett centralt begrepp i detta sammanhang är *intentionalitet*. Intentionalitetsbegreppet låter sig inte med enkelhet beskrivas eller diskuteras på några få rader, vilket ger anledning till att lyfta fram det som ett av fenomenologins mest centrala begrepp i studiens teorikapitel.⁴ Övergripande kan intentionalitet beskrivas som att medvetandet alltid är riktat mot något annat än sig självt. När vi tänker, känner, handlar etc. är det alltid något vi riktar uppmärksamheten mot och att det som erfars, alltid erfars som något, det har en mening för den som erfär. I denna studie är det elevernas beskrivningar, av vad som konstituerar sig i deras livsvärld när de beskriver sin programmeringshandling, som är föremål för fenomenografisk analys.

Intresset för programmering har bland annat sin utgångspunkt i mina erfarenheter som lärare i programmering vid Karlstads universitet. Under mina sejourer i undervisningen har jag ofta funderat på varför programmering upplevs krångligt för många studenter. Vad är det som är komplicerat och hur tänker studenterna när de programmerar? Klassner och Anderson (2003) pekar på att det är svårt för studenterna att se kopplingen mellan programmeringsmomentet och resultatet. Kindborg (2003, s. 49) resonerar på liknande sätt och uttrycker det så här:

While programming can be fun and creative, learning how to program can be difficult, both for children and for adults. One reason for that programming can be hard is that there exists a mental distance, a kind of gap, between the program (the source code) and the result seen on the computer screen when running the program.

Klassner och Anderson (2003) och Kindborgs (2003) framställning av glappet mellan programmet och resultatet är en vanlig beskrivning av var i processen lärandeproblematiken kan uppstå. Jag vill redan i detta inledningskapitel

³ Theman (1983): Psykoanalysen. Kroksmark (1987): Livsvärldsfenomenologin. Dahlin (1989): Hermeneutiken.

⁴ Se kapitel 4.

behandla några aspekter av detta för att tydliggöra var föreliggande studie positionerar sig när det gäller problemområdet. Nämda författares beskrivningar har problemfokus på tekniken och den distans som uppstår i relationen mellan olika moment inom datorn, källkoden och skärmen där programmet körs. Jag menar att denna bild är allt för tekniskt fokuserad där människans erfarenheter av programmering hamnar i skymundan. För att möta detta är föreliggande studie inriktad mot hur eleverna själva definierar och beskriver programmering med fokus på den subjektivt erfarna programmeringshandlingen. Mina erfarenheter är att programmeringsundervisningen tenderar att reduceras till lärande av specifika kommandon (syntax) och de krav på design och struktur som följer med respektive programspråk. Det vill säga, att fokus ligger på att lära ett specifikt programspråk istället för att lära sig programmering som arbetsmetod och det tankesätt som utgör grunden för att kunna programmering generellt oavsett vilket programspråk vi står inför.

Funderingarna kring lärande av programmering hamnade i nytt ljus i samband med att jag deltog i forskningsprojektet *Programmerbart konstruktionsmaterial i undervisningen*⁵ (Lindh m.fl, 2003). Syftet med projektet var bland annat att studera vilka färdigheter eleverna utvecklade då de arbetade med programmerbart konstruktionsmaterial i undervisningen. Eleverna skulle bygga en robot som sedan programmerades för att utföra uppgifter. För att möta Klassner och Anderson (2003) och Kindborgs (2003) synpunkter blir kopplingen mellan programmering och resultat tydligt med det programmerbara konstruktionsmaterialet eftersom resultatet är ett erfalande av ett fysiskt objekt - roboten. Robotens rörelser är/var en direkt återkoppling till sekvenserna i programmet. Flera elever visade att goda programmeringskunskaper utvecklades under perioden som projektet pågick. Arbetet var kopplat till teknikundervisningen där målet var att eleverna skulle utveckla kunskaper om och bli mer intresserade av teknik genom att konstruera och programmera robotar som skulle utföra olika uppgifter. Exempelvis att låta roboten åka över en bro för att hämta något föremål som sedan skulle föras tillbaka till utgångsläget. Som en del i projektet intervjuades elever där de beskrev funktionen och uppbyggnaden av ett program som de själva konstruerat. Ett viktigt resultat var att elevernas beskrivningar av programmen utgick från roboten och det uppdrag som skulle utföras. Resultatet visade även att eleverna inte upplevde några specifika svårigheter

⁵ *Programmerbart konstruktionsmaterial i undervisningen*. Ett samarbete mellan Karlstads universitet och Högskolan i Jönköping. I projektet följdes 13 klasser i Mellansverige i årskurserna 4-5 och 8-9 under perioden mars 2002 till maj 2003. Syftet var bland annat att undersöka vilka färdigheter eleverna utvecklade då de arbetade med programmerbart konstruktionsmaterial i undervisningen.

INLEDNING

med programmeringen (Segolsson, 2005). De flesta eleverna lärde sig momentet utan större svårigheter, mycket tack vare programmeringsmiljöns enkla uppbyggnad och att eleverna fick resultatet via ett fysiskt objekt och inte via en applikation i datorn. Likväl visade det sig att flera elever önskade få mer hjälp från lärarna, så att de kunde utveckla sina programmeringskunskaper i syfte att kunna göra mer avancerade program. Detta var en central fråga under projektet. Både elever och lärare var eniga om att lärarens kunskaper var avgörande för hur väl eleverna lärde sig programmering. Lärarna menade att de saknade erfarenhet av denna typ av arbete och att de själva kände sig osäkra inför programmeringsmomentet.⁶ Flera lärare efterlyste utbildning och handledning i hur de ska möta eleverna i programmeringsundervisningen. Erfarenheterna från projektet ledde till att jag ville studera vad eleverna själva uppfattade som centrala aspekter under programmeringshandlingen och med utgångspunkt från detta skapa en ökad insikt i hur lärandet går till.

För att besvara mina frågor fördjupades studien kring det programmerbara konstruktionsmaterialet genom ett projekt som startade hösten 2003, *Lärande och programmerbart konstruktionsmaterial*.⁷ Ett av projektets huvudsyfte var att göra en processnära forskning av hur eleverna begreppsbildar när de arbetade med programmering för att kunna förstå hur de lär sig momentet. Genom att videofilma applikationen (programmeringsmiljön) när eleverna programmerade för att sedan använda detta som utgångspunkt vid intervjuerna söktes elevernas intentionala objekt i deras beskrivningar av programmeringshandlingen. Det är intervjudata från detta projekt som ligger till grund för att besvara forskningsfrågan i föreliggande studie. Urvalsgruppen är elever från skolår 8 och 9 vid tre skolor i mellersta Sverige.

Forskningsansatsen i föreliggande studie skiljer sig från vad som gjorts tidigare inom området. Utbildning och forskning om programmering är nästan uteslutande kopplade till ämnen inom den naturvetenskapliga och samhällsvetenskapliga disciplinen där teknisk funktionalitet och effektivitet är rådande forskningsintressen. Resultaten bygger i stor utsträckning på hypotesprövning, där diskussioner om hög teknisk prestanda sätts före människans erfarenheter

⁶ Här fanns några undantag. Två av lärarna var vana med både datorarbete och programmering, i dessa klasser var inte problematiken lika påtaglig.

⁷ *Lärande och programmerbart konstruktionsmaterial*. Ett samarbete mellan Karlstads universitet och Högskolan för lärande och kommunikation i Jönköping. Projektet pågick under 2003 – 2004.

av datoranvändning. Detta är något som skjuter vid sidan av målet enligt Wallén (1993):

Människor handlar efter sin uppfattning – inte efter hur det faktiskt förhåller sig eller hur det borde vara (ur någon rationalitetsaspekt) [...] Allmänt gäller för teknik (liksom sociala organisationer) att den inte kan bedömas med avseende på prestationsförmåga och effektivitet utan att detta sätts i relation till hur den används av brukaren. Detta är tydligt vid datoranvändning. (Wallen, 1993, s. 68)

Nödvändigheten av ett processnära individperspektiv när det gäller datoranvändning i skolan är något som även Becker (2000) efterlyser. I en debattartikel i tidskriften *Datorn i utbildningen* poängterar han vikten med att komplettera den övergripande bild av datorarbete i skolan med närgångna och lokala analyser där man på nära håll följer konkreta undervisningsprocesser.⁸ Programmering är en aspekt av datoranvändning och jag menar att ett processnära individperspektiv är nödvändigt när det gäller forskning om programmering i skolsammanhang för att förstå hur detta lärande går till. I denna processnära studie står den lärande elevens konkreta uppfattningar av programmering i centrum.

Studien är inte enbart viktigt för utvecklingen inom programmeringsundervisningen, utan bidrar även till ett ontologiskt och metodiskt sätt att angripa frågeställningar inom ett växande forskningsområde – lärande och IT.

UTGÅNGSPUNKTER I FRÅGESTÄLLNINGEN

Utan att föregripa innehållet i teorikapitlet, vilket givetvis utgör studiens huvudsakliga utgångspunkter, vill jag innan forskningsfrågan presenteras peka på några viktiga aspekter som berör denna.

Ur ett lärandeperspektiv är det enligt Marton och Booth (2000) viktigt att förstå hur man erfar lärandets objekt – det objekt som är föremål för lärandet – därför att själva erfandets innehåll avgör inlärningsresultatet på grundval av tidigare erfarenheter hos den lärande. På en allmän nivå är beskrivningen av lärandeobjektet i denna studie *lärande av programmering* och det specifika studieobjektet

⁸ Becker använder i artikeln metaforen flygfoto över vägarna: "... Det är förstas inget fel på flygfoton i sig. Felet är om man tror sig kunna utvärdera vägnätets transportduglighet utan att landa. Det räcker heller inte att landa och titta. Man måste ut och provköra vägarna. Mängden av vägar kan förstas vara avskräckande. En metod är då att komplettera flygfotots övergripande bild med vissa utvalda landningsplaster där vägar provkörs. Och provkörs under olika trafik-, ljus- och väderförhållanden. Tillsammans kan dessa metoder ge både en övergripande bild och en detaljerad förstelse av hur vägnätet duger för transporter." (2000, Nr. 7, Debatt)

INLEDNING

utgör en aspekt av detta lärande. Avsikten är att redogöra för vad eleverna beskriver att de uppfattar som centralt framträdande under programmeringen. Detta innebär att uppfattningsbegreppet är viktigt för studiens vidkommande där ett särskilt fokus är att beskriva mot vad uppmärksamheten är riktad i elevernas beskrivningar. Objektet som eleverna beskriver under intervjuerna är den egna programmeringshandlingen där uppfattningarna utgörs av elevernas förståelse av sitt arbete.

Studieobjekts centrala kärna, uppmärksamhetens riktadhet, beskrivs bäst – som tidigare nämnts – genom det fenomenologiska begreppet *intentionalitet*.⁹ Andra centrala begrepp för studieobjektet är *handling*¹⁰ och *programmering*.¹¹ Det är handlingen programmering som utgör objektet i studien. Det vill säga själva akten då ett program konstrueras i den programmeringsmiljö som eleverna har till förfogande. För att kunna närma mig studieobjektet och det kunskapsanspråk som kan göras i detta sammanhang tar jag hjälp av fenomenografins *uppfattningsbegrepp*.¹² I förhållande till kunskapsbegreppet får detta en speciell innebörd. Kroksmark (1987) använder begreppet uppfattning för att ge en utvidgad innebörd av kunskapsbegreppet. Han menar att:

[...] kunskap betyder då att omfatta eller få insikt i och implicit eller explicit förstå något upplevt, iakttaget eller erfaret. Kunskapen är relaterad till tänkandets intentionalitet där intentionaliteten omfattar en viss innebörd av det vi har kunskap om. (Kroksmark, 1987, s. 228)

En uppfattning integrerar dessa skilda aspekter av kunskapsbegreppet och att begreppet kan utnyttjas ”för att ge innebörd åt substantivet kunskap i betydelsen resultat” (Ibid, s. 228). En uppfattning utgör en avgränsning av hur ett innehåll behandlas, vilket för med sig möjligheten att visa de delar ”i den intentionala akten som leder subjektet fram till att en viss innebörd konstitueras” (Ibid, s. 233). Min tolkning är att uppfattningsbegreppet i detta sammanhang innefattar programmeringshandlingen som tankeakt, det sammanhang i vilket denna handling sker och resultatet av handlingen. Med resultatet av handlingen menar jag det färdiga programmet.

⁹ Se kapitel 4.

¹⁰ Se kapitel 4.

¹¹ Se kapitel 3.

¹² Se kapitel 5.

FORSKNINGSFRÅGAN

Mot tidigare presenterad bakgrund och utgångspunkter för frågeställningen avser jag att söka svar på hur eleverna uppfattar sin egen programmeringshandling och vad eleverna riktar sin uppmärksamhet mot i sina beskrivningar. För att studera detta tar jag hjälp av fenomenologins intentionalitetsbegrepp och studiens fokus är riktat mot de intentionala objekt som framträder då eleverna beskriver sina programmeringshandlingar. Genom att tolka utsagorna av elevernas uppfattningar försöker jag med andra ord förstå *vad* det är som framträder och *hur* detta framträder för eleverna. Forskningsfrågan som ligger till grund för resultatet är:

- Vad riktar eleverna sin uppmärksamhet mot i sina beskrivningar av programmeringshandlingen?

Som tidigare nämndes beskrivs den riktadhet hos eleverna som är föremål i studien *bäst* genom intentionalitetsbegreppet. Anledningen till denna försiktighet är att jag inte gör anspråk på annat än att beskriva det som framträder genom elevernas utsagor. Det är svårt, om inte omöjligt, att till fullo beskriva en annan persons intentionala objekt. Det är också mina tolkningar av elevernas beskrivningar av samspelet mellan deras handling och programmeringsresultat som ligger till grund för resultatet. Det är med utgångspunkt i detta som jag vill betona att det är uppfattningar av programmeringshandlingen som är i fokus i denna studie och att intentionalitetsteorin bidrar till att intervjuerna analyserats med hänsyn till vad eleverna avgränsar som framträdande objekt i sina beskrivningar.

SYFTE

Uppsatsens överordnade syfte är att förstå och beskriva vad eleverna riktar sin uppmärksamhet mot under programmeringshandlingen. I enlighet med den fenomenografiska teoribildningen utgår jag från att eleverna som programmerar gör detta på kvalitativt olika sätt. Metodiskt innebär detta att förstå och beskriva hur variationen av uppfattningar mellan eleverna och hos eleverna ser ut (Marton & Booth, 2000). Avsikten är att belysa en aspekt av programmering som kan läggas till annan kunskapsbildning för att utveckla ny kunskap inom detta område.

I ett vidare syfte vill jag, inspirerad av bland andra von Wright (1986) och Winograd och Flores (1988), medverka till utvecklingen av ett erfarenhetsfilosofiskt förhållningssätt inom den teknologiska disciplinen. Med utgångspunkt i Heideggers filosofi för Winograd och Flores fram att deras ambition är att försöka förstå hur människor använder datorer. För att nå denna förståelse framhäver de vikten av att designers och användare av datorsystem måste ta utgångspunkt i utfallsrummet av möjliga *breakdowns*. von Wright menar att människans erfarenhet, föreningen av vetande och kunnande, är grunden för allt det vi idag kallar teknik. Åsidosätter vi människans uppfattningar av tekniska objekt, riskerar vi att underbygga en deterministisk samhällssyn. Med stöd i livsvärldsfilosofin vill jag bidra till erfarenhetsperspektiv där elevernas erfarenheter av att programmera görs till utgångspunkt för diskussioner om undervisning och lärande av programmering.

POSITIONERING

När studien genomfördes var jag anställd vid ämnet informatik vid Karlstads universitet samt antagen som licentiand i pedagogiskt arbete vid samma lärosäte. Under fem års tid har jag haft min huvudsakliga undervisning inom informatik och lärarutbildningen. Det var genom undervisning och handledning i flertalet kurser i programmering som jag började fundera över de svårigheter som många studenter hade när de skulle lära sig programmering. Funderingarna aktualiserades på nytt i samband med nämnda studier med det programmerbara konstruktionsmaterialet. Genom min lärarutbildning, och sedermera undervisning av blivande lärare, väcktes mitt intresse för pedagogik och filosofi. 2003 blev jag antagen till forskarutbildningen och jag startade mina studier med stor entusiasm dels för jag fick möjligheten att forska och dels för att jag kunde göra detta med ett flervetenskapligt perspektiv. Jag stod med ena foten inom teknisk disciplin och med den andra inom utbildningsvetenskaplig disciplin.

Pedagogiskt arbete är ett forskningsområde med koppling till universitetens och högskolornas yrkesinriktade utbildningar och till pedagogisk yrkesverksamhet som sker inom och utanför utbildningssystemet. Forskningen ska bland annat bidra till kunskap och teoribildning om den lärande människan med utgångspunkt i pedagogisk handling, där teori och praktik behandlas ömsesidigt beroende av varandra. Vidare är ”en grundläggande del av forskning i pedagogiskt arbete är att utveckla vetenskapliga teorier och forskningsmetoder som är handlingsgenererande” (Kroksmark, 2002). Det är praktisknära frågor med utgångspunkt i pedagogiska frågeställningar i olika aktörers situationer som

är centrala inom pedagogiskt arbete. I föreliggande studie är forskningsobjektet hämtat från skolans praktiska verksamhet och min ambition är att resultatet ska berika undervisningen av programmering.

BESKRIVNING AV KONSTRUKTIONSMATERIALET

Studien tar sin utgångspunkt i situationen där en elev fått i uppgift att programmera en robot för att lösa ett specifikt uppdrag. Robotarna är konstruktioner av teknik-Lego¹³ som till stora delar utgörs av traditionellt Lego men med viss teknifiering. Bland annat ingår en programmerbar enhet som lagrar programmen och styr roboten. Dessutom finns sensorer av olika slag, kuggjul, motorer och en mängd andra detaljer som är inspirerade av vårt tekniska samhälle. När en robot konstrueras utgår eleverna – i de flesta fall – från den programmerbara enheten, den utgör själva kroppen och det är till den som motorer och sensorer kopplas via sladdar. I materialet ingår dessutom en programvara som möjliggör att robotkonstruktionerna kan programmeras. När eleven programmerat färdigt laddas programmet över från datorn till roboten via en sensor. Programmeringsmiljön är grafisk och innebär att eleverna använder ikoner som sätts samman till ett program (Preece, 1994). Varje ikon representerar en programfunktion och de placeras i den ordning som roboten ska utföra kommandona. Programmeringsmiljön (RoboLab 2.01) vänder sig till såväl nybörjare som expert eftersom möjligheten finns att konstruera både enkla och avancerade program. I bilden på nästa sida (Bild 1) visas den robot som eleverna programmerade i denna studie.

¹³ Namnet på materialet är LEGO Mindstroms. En tidigare version som förekom i forskning under 1980 och 1990-talet hette LEGO Dacta.

INLEDNING

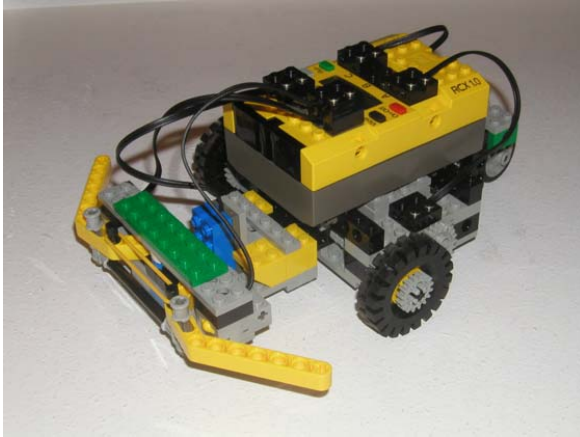


Bild 1. Roboten som eleverna programmerade i studien.

Det område som roboten rörde sig inom och där uppgifterna skulle utföras var ett område om cirka två gånger tre meter och kallades allmänt för banan. Banan består bland annat av hus, broar, en väderkvarn, staplade tunnor och andra ting byggda av Lego. Allt är uppställt likt ett landskap där underlaget utgörs av svarta linjer på en vit grund. Kontrasten mellan de svarta linjerna mot det vita underlaget möjliggör för roboten att orientera sig med hjälp av en ljussensor. I bilden nedan visas ett exempel på hur delar av en bana (uppgiftsområde) kan se ut.

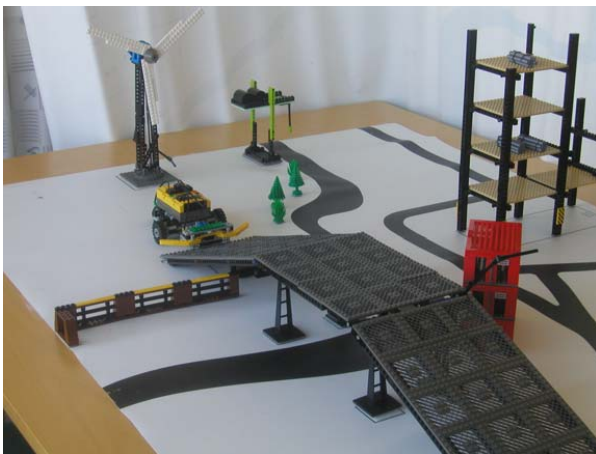


Bild 2. Delar av en bana där roboten ska utföra olika uppgdrag.

INLEDNING

Uppgifterna kunde exempelvis utgöras av att roboten skulle hämta eller flytta något, att förflytta sig för att stänga en broklaff eller att starta en väderkvarn. I de fall där eleverna tävlade mot varandra skulle uppdragen utföras på tid. Tidsaspekten och att de olika uppdragen gav olika poäng medförde att det låg mycket taktik bakom elevernas val av vilka uppgifter som skulle lösas innan tiden var slut.

Materialet har framgångsrikt använts i teknikundervisningen (Lindh m.fl, 2003) eftersom det erbjuder stor variation och eleverna får möjlighet att arbeta praktiskt med problemlösning och tekniska komponenter. En tidigare studie har visat att eleverna bland annat får förståelse för hållfasthet och hur utväxling fungerar. Dessutom har arbetet varit ett bra sätt att introducera datoranvändning i skolan och en viss ökning av en positiv attityd till teknik har också påvisats (Lindh m.fl, 2003). Ett annat område där konstruktionsmaterialet används är inom forskning kring artificiell intelligens. Robotar programmeras då för att ingå i en sluten grupp (Community) för att sedan lära sig rörelsemönster så att de inte stöter mot någon annan robot eller annat hinder (Nordin & Wilde, 2003).

2. TIDIGARE FORSKNING

För att beskriva forskning om programmering och lärande med programmering finns det skäl att göra ett vidare avstamp för att placera studien i ett större sammanhang. Forskning om lärande har nästan uteslutande varit knuten till den pedagogiska eller utbildningsvetenskapliga disciplinen, medan programmering har en hemvist inom forskningsintressen som tar sin utgångspunkt i datavetenskap eller informatik. Programmering ingår i ett sammanhang som har med datorer att göra, som i sin tur har med användningen av dem att göra på något sätt. Den vetenskap inom vilken man forskat om att utveckla tillämpningar för användarna är informatik. Sedan 1960-talet har den huvudsakliga inriktningen inom informatiken varit att studera utvecklingsprocessen för tillämpad datoranvändning.

Även om programmering ingår i informatikutbildningarna är forskningen högst ringa när det gäller hur vi lär oss programmering eller vad användarna uppfattar som centrala aspekter när de programmerar. En anledning till detta är att intresset för programmeringsforskningen i första hand varit inriktad mot den tekniska utvecklingen och inte de pedagogiska aspekterna eller undervisningen om programmering. Emellertid har några forskare aktualiserat frågor om lärande för att förbättra anpassningen av datorsystemets tillämpningar. Høyer (1979) hävdar exempelvis att systemutvecklingsprocessen bör bygga på en pedagogisk teori. Hägerfors (1995) betonar betydelsen av att kunna samlära i gruppen för att resultatet av systemutveckling ska bli framgångsrikt. Carlsson (2000) behandlar bland annat didaktiska frågor för att förbättra lärandet när systemutvecklingens olika aktörer kommunicerar.

Utveckling inom informatikforskningen går mot ett synsätt där människan sätts i centrum för att bra IT-tillämpningar ska kunna utvecklas. Här har en tradition inom Human Computer Interaction (HCI) vuxit fram som en företrädare för människans betydelse när det gäller IT-frågor. HCI utgår nästan uteslutande från kognitiv psykologi (jfr. Preece, 1994; Dix m.fl., 1998) där huvudidén är att utveckla tillämpningar med hänsyn till människans kognitiva förmåga och då helst göra tillämpningarna så enkla att de inte behövs någon utbildning för att använda dem. Ett annat exempel på hur det mänskliga perspektivet lyfts fram

när det gäller IT är Lindblad-Gidlund (2005), som behandlar ett relationsperspektiv mellan människan och informationsteknologi. Hon diskuterar konsekvenser, förutsättningar och möjligheter i denna relation ur ett socialkonstruktivistiskt perspektiv inspirerat av Berger och Luckman (1967). Centrala problemområden för Lindblad-Gidlund är ”known and unknown views on technology” och den avgörande frågan ”what is technology to you?” (s. 88) ställs i samband med hennes ambition att presentera en metod för att medvetengöra relationen mellan människa och teknologi.

Mot ovanstående skissartade överblick kan det konstateras att flera företrädare inom det informationsteknologiska området pekar på att det behövs pedagogiska teorier inom den tekniska disciplinen. Det är i spåret av denna forskning som jag gör mitt avstamp för att hitta nya nyanser inom området programmering. Studiens forskningsfråga har sin grund främst inom två forskningsområden: Forskning om lärande med konstruktionsmaterialet och forskning om lärande och programmering. Nedan görs en redogörelse för dessa områden i syfte att beskriva de riktningar av tidigare forskning som präglar denna studie.

LÄRANDE MED KONSTRUKTIONSMATERIAL

En av inspiratörerna till att införa programmerbart Lego-material i skolan var Seymour Papert. Han var en av initiativtagarna till det uppmärksammade projektet INFOESCUELA (Papert, m.fl, 1999). Projektet genomfördes i Peru med syftet att studera de pedagogiska effekterna, materialets inverkan på skolkulturen liksom frågor om attityder till materialet. I huvudsak gick projektet ut på att jämföra två grupper av elever. Den ena gruppen arbetade med konstruktionsmaterialet – de exponerades för material som det heter i många hypotesprövande studier – och den andra gruppen av elever använde inte materialet men de gjorde samma typ av tester. Resultatet visade att det fanns signifikanta skillnader mellan experimentgrupp och kontrollgrupp främst när det gällde att lära sig matematik och teknologi. Resultatet visar också att materialet uppmuntrar till logiskt tänkande, egen kreativitet och fantasi samt att det var motivationshöjande. Det finns som vi senare ska se en del kritik mot Paperts teorier.

Teoretiskt byggdes projektet på Paperts tankar om ”Konstruktionism” (Papert m.fl., 1999, s. 37), en benämning för ett teoretiskt resonemang som förklaras med hjälp av nedanstående punkter:

- Learning happens especially well when children are engaged in constructing a meaningful product, such as a sand castle, a poem, a machine, a story, a piece of artwork, a computer program, or a song.
- When children are involved in creating something, making something, building something, they are simultaneously building knowledge in their minds. They are trying out ideas, making conjectures and testing them, making connections between ideas or reorganizing them – in short, they are building knowledge structures.
- This newly formed knowledge enables children to build even more sophisticated constructions, which yield still more knowledge...and so on, in a self-reinforcing cycle.

Om vi skulle ha låtit Dewey (1998) ha sammanfattat punkterna ovan skulle han med största sannolikhet ha definierat detta som *learning by doing*. Det är också ur den pragmatiska utgångspunkten som vi kan se att Paperts tankar ofta sammanfaller med Deweys.

I Sverige genomfördes åren 2002-2003 en liknande studie som den peruanska genom projektet *Programmerbart konstruktionsmaterial i undervisningen* (Lindh m.fl, 2003). Projektet var till stora delar designat på samma sätt som INFOESCUELA. Hypotetiskt antogs att arbetet med materialet skulle ge effekter på både färdigheter inom matematik och teknik, och att attityden till teknik i allmänhet skulle öka. Resultatet visade att det inte fanns samma skillnad i den svenska studien beträffande matematik jämfört med den peruanska. Emellertid visades en viss ökning när det gällde attityden till att arbeta med tekniskt material i undervisningen. Även en viss ökning när det gäller kunskaper i teknikämnet kunde påvisas i de grupper som arbetat med materialet jämfört med kontrollgrupperna. Resultatet bygger på intervjuer och enkäter där klasser som arbetade med materialet jämfördes med kontrollklasser som inte använde materialet.

Idéerna om att använda grafiska programmeringsgränssnitt för att elever redan i unga år skulle lära sig att programmera datorer förde Papert fram i början av 1980-talet.

Genom det grafiska programspråket Logo ville Papert studera hur elever i unga år klarade av att programmera datorer och som ett resultat av detta arbete skulle eleverna utvecklas i deras kreativa problemlösningsförmåga (Papert, 1994; 1999). Papert hävdade att genom programmeringen skulle eleverna dels lära sig att hantera datorn och dels bli bättre i problemlösning. Visionen var att datorn kunde vara ett redskap för det goda lärandet och en förutsättning för en slags alternativ pedagogik där det fria arbetet var den röda tråden. De som ifrågasatte Paperts idé om det fria arbetet menade att denna vision stannade vid att vara en idealbild. Lilja och Lindström (2002) anser att Paperts idéer bör hanteras kritiskt. Med referens till Kurland m.fl (1987) menar Lilja och Lindström att det främst är förväntningarna på att barnen skulle lära sig att programmera genom att helt fritt utforska datorn som måste kritiseras. En uppföljningsstudie där Paperts teser testades visade att helt elevstyrd verksamhet som pedagogisk metod ”för att lära ut programmering inte var så effektiv som man ansåg att Papert hade menat” (Ibid, s. 36). Papert replikerade på kritiken genom att påpeka att arbetet inte var tänkt att vara helt fritt. Läraren ska arbeta med eleven och viss stimulans och provokation är nödvändig för att eleven ska uppnå resultat.

Lilja och Lindström (2002) tar i sin undersökning utgångspunkt i det kritiska förhållningssätt till *det fria arbetet* som Papert förde fram och som belystes ovan. I deras studie ställs frågan om vad eleverna under fria förutsättningar gör med Lego-Dacta (se not 13) och vad som karakterisera elevernas arbete? Teoretiskt bygger Lilja och Lindströms studie på sociokulturella antaganden och de hävdar i denna anda att:

[...] Lego-Dacta erbjuder en miljö som stödjer intressanta former av social interaktion där problemlösning kan bli ett kollektivt fokus, och där arbetet kan organiseras och fördelas på olika sätt. Diskussioner om problemen eleverna stöter på i miljön kan leda till fruktbara perspektivskiften och ge upphov till nya sätt att tänka gemensamt. (Lilja & Lindström, 2002, s. 54)

Via citatet kan vi förstå att författarna ser vinster med materialet när det gäller det kollektiva lärandet och att *det fria arbetet* ger gemensamma effekter i gruppen som arbetar med materialet. Deras fortsatta argumentering poängterar emellertid att arbetet med exempelvis programmering har en sluten och konkret prägel och att det inte relaterar till andra kunskapsområden utanför situationen, vilket Papert hävdade.

Trots att Lilja och Lindström (2002) för fram att arbetet inte kan göra anspråk på några transferkunskaper, och att de riktar kritik mot Paperts visionära tankar om att den moderna tekniken skulle ersätta skolan och de traditionella undervisningsformerna, så konstaterar författarna att:

Modern informations- och kommunikationsteknik, som Lego-Dacta är ett exempel på, är verktyg som kan förstärka olika sidor av en existerande praktik och det är ur praktikens synpunkt viktigt att ha en klar uppfattning vilka de är. (Lilja & Lindström, 2002, s. 56)

Med samma forskningsföremål, Lego-Dacta, visar Ivarssons (2002) resultat att eleverna har svårigheter med att lära sig programmeringsmomentet på en djupare nivå vid denna typ av arbete. Nedan beskrivs ett moment hämtat från Ivarssons studie där en elev ska lägga in en ny instruktion i ett befintligt program, som denne redan vet fungerar:

De försöker nu lägga in vad de kallar ”stopp” (dvs. ikonen med den öppna strömbrytaren) i den del av koden som de utifrån den tidigare diskussionen vet kommer exekveras. Målet är att stoppa *proceduren* (deras skrivna programkod). [...] Tillvägagångssättet och lösningen har mycket karaktär av trial-and-error och innehåller inget kvalitativt nytt. De lyckas inte på detta stadium resonera sig fram till att problemet har en logik som går utanför vad de tidigare lärt sig i denna miljö, utan de arbetar fortfarande med de ramar som etablerats under det gångna passet. (Ivarsson, 2002, s.69)

Studien bygger på att elever från skolår 6 ska lära sig rekursion¹⁴ med hjälp av Dacta-materialet och programmeringsmiljön. Ivarsson hävdar att den grafiska programmeringen till viss del hjälper till när eleverna ska beskriva hur de förstår rekursionsproblemet eftersom de kan peka samtidigt som de beskriver. Likväl gäller att eleverna använder uttryck isolerade till programmeringsmiljön och de har svårt att tillägna sig en förståelse för problemets art i en vidare mening. Ett exempel på detta är att lägga till ett nytt moment i det befintliga programmet som citatet ovan visar. Då eleverna ska göra detta upplevs problemet vara något helt nytt, de börjar om från början med hela problemlösningen och provar sig fram till en ny lösning. Det är svårt att dra några egentliga slutsatser kopplade

¹⁴ Rekursion: En programfunktion som anropar sig själv tills ett villkor är uppfyllt. Då resultatet presenteras är det ett antal kopior av funktionen som visas, dock kan kopiorna innehålla andra värden än ursprungsfunktionen. Marton och Booth (2000, s.94) ger oss ett exempel från vardagen på rekursion: ”[...] när du tittar i en stor väggspegel, när det finns en annan väggspegel lite snett bakom dig, något som de flesta kläddaffärer fiffigt nog erbjuder. Om du tittar in i spegeln rakt framför dig, kan du se en reflektion av spegeln bakom dig, och åtminstone en del av din rygg där. Bakom dig kan du också se en återspeglning av spegeln framför dig [...] Du ser i princip ett oändligt antal kopior av din rygg och ditt ansikte [...] Detta är på ett sätt ett rekursivt fenomen, i det att det finns ett stort antal kopior av samma sak [...]”

till förståelsen av programmeringsmomentet i Ivarssons (2002) studie. För det första är problemets art, att förstå rekursion, så avancerat att frågan är om det är rimligt att en elev i skolår 6 kan tillägna sig förståelse på denna abstraktionsnivå. För det andra blir konsekvensen av detta att det är omöjligt att veta om det är programmeringen eller det matematiska problemet som eleverna har svårigheter med. Ivarsson hävdar också, att problemställningen vanligtvis hör hemma på universitetsnivå och att ”i förhållande till elevernas utbildningsnivå kan problemet ses som mycket avancerat, vilket gör händelseförloppet särskilt intressant att undersöka” (Ibid, s. 61).

Den tidigare forskning som rört konstruktionsmaterialet har företrädesvis sin utgångspunkt i sociokulturella teorier. I ovanstående redogörelse märks detta framförallt hos Lilja och Lindström (2002) och Ivarsson (2002). Det går även att se samma teoretiska ansats hos Papert även om den inte är lika tydligt uttalad. Emellertid blir det konstruktivistiska synsättet framträdande i Paperts forskning, vilket tydliggörs genom hans egen teoretiska benämning: Konstruktionism. När det gäller de teoretiska utgångspunkterna skiljer sig föreliggande studie från den beskrivna dels genom att teorigrunden utgår från människans erfarenheter av den konkreta upplevda världen och dels genom att kunskap ses som något som konstitueras, inte konstrueras.¹⁵

Gemensamma nämnare mellan denna studie och de som lyfts fram ovan finner vi i det programmerbara konstruktionsmaterialet, där grafisk programmering används som tillvägagångssätt för att programmera. Här finns det likheter med Ivarssons (2002) forskning, därför att den är inriktad på elevernas förståelse av rekursion vilket kan ses som en aspekt av programmeringshandlingen. Emellertid skiljer sig forskningsobjektet åt i föreliggande studie jämfört med Ivarssons genom att forskningsfrågorna har olika inriktningar och intressen samt att det är äldre elever som ligger till grund för denna undersökning.

I min sökning av tidigare forskning har jag inte påträffat några studier som behandlar vad eleverna riktar sin uppmärksamhet mot när de beskriver sina programmeringshandlingar. Föreliggande studie utgör alltså ett perspektiv som är viktigt att lägga till tidigare forskning inom området för att bättre förstå hur programmering erfars av eleverna och vad som är centrala aspekter att utgå från när det gäller programmeringsundervisning i dessa sammanhang.

¹⁵ Se kapitel 4.

LÄRANDE OCH PROGRAMMERING

När det gäller metodisk ansats och intresset för uppfattningar av programmering finns det likheter med föreliggande studie och den som presenterades av Booth (1992). Booth redovisar hur universitetsstudenter erfar ett rekursionsproblem, som ska lösas med hjälp av programmering. Likheterna mellan studierna finns dels i att båda undersökningarna behandlar frågan om hur en specifik urvalsgrupp uppfattar att de lär sig programmering och dels i att den metodiska ansatsen är fenomenografi i båda studierna. I delar av Booths resultat presenteras fyra kvalitativt skilda beskrivningskategorier av hur programmering av rekursionsproblemet uppfattas. De två första kategorierna visar att studenterna är fokuserade på programmeringsspråket eller själva syntaxen. Kategori tre och fyra pekar på något bortom själva instruktionerna till datorn, dessa har fokus på hur problemet skulle lösas med hjälp av programmeringen. I figuren nedan redovisas en sammanfattning av Booths (1992) kategorier:

- A. **Learning to program as learning a programming language**, in which focus is on learning the features and the details of one or more programming languages;
- B. **Learning to program as learning to write programs in a programming language** in which focus is on learning to write programs which make use of available techniques and features of the programming language;
- C. **Learning to program as learning to solve problems in the form of programs**, in which focus is on learning to produce programs according to the needs of problems;
- D. **Learning to program as becoming part of the programming community**, in which focus is on learning to solve problems and write programs in collaboration with, or for, someone else, and thereby participate in the world of programming.

Figur 1. Booths fyra kategorier av kvalitativt skilda sätt som studenter erfar programmering av ett rekursionsproblem. Kategorierna är delar av resultatet.

I en senare redovisning av samma studie hävdar Marton och Booth (2000) att lärarnas uppfattning av det centrala undervisningsbegreppet rekursion inte hänger samman med hur studenterna uppfattade det. Lärarnas undervisningsmetoder missgynnar vad som krävs för att programmera och för att eleverna ska kunna lära sig programmera.

Den frågeställning som står i fokus i Berglunds (2005) avhandling är ”hur kan man på pedagogisk vetenskaplig grund förstå studenternas lärande av datorsystem i en komplex läromiljö?” (s. 219). Studien gick ut på att förstå hur studenterna lärde sig att lösa datakommunikationsproblem i ett system som de gemensamt konstruerade. Likheterna mellan föreliggande studie och Berglunds ligger i att båda studerar förståelse av problem knutna till datorer. Berglunds ansats har emellertid en stark datavetenskaplig prägel eftersom det är val av nätverksprotokoll och digitala kommunikationsregler som står för lärandeobjektet. För att förstå hur studenterna erfor lärandet i detta sammanhang, valde Berglund att analytiskt separera uppfattningar av vad, hur, varför och var studenterna lär. Metodiskt använder sig Berglund av fenomenografi – på ett innovativt sätt – där han låter de metodiska utgångspunkterna integreras med element ur verksamhetsteorin. Tydligast syns denna sammanfogning i hans diskussion kring kontexten, en diskussion som bygger på resultatet från frågan *var studenterna lär*. Berglund (2005, s. 220) menar:

För att möjliggöra den syntes av fenomenografiska data, som erbjuds av verksamhetsteorin, har frågan om sammanhang, eller kontext, till lärandet belysts ur olika perspektiv. En åtskillnad gjordes på basis av vem som erfar en viss kontext i en viss situation. En student erfar en kontext som en bakgrund till det fenomen som diskuteras i en intervjusituation.

Berglunds ansats är inte oproblematiserad. En central utgångspunkt inom fenomenografien är att informantens beskrivningar lyfts ur sitt tidsmässiga och rumsliga sammanhang. Med motivet att man på det sättet kan föra ihop resultat från olika sammanhang och vice versa blir ett resultat användbart i andra kontexter än den som studerats (Marton & Booth, 2000). Fenomenografien har heller inget intresse i vem som säger vad, eftersom det är innehållet i utsagorna som ligger till grund för analysen. I Berglunds fall är kontexten (var studenterna lär) ett forskningsobjekt, det vill säga intresset är riktat mot uppfattningar av kontexten, vilket egentligen inte möter något hinder. Likväl medför föreningen med verksamhetsteorin, som tydligt återspeglas i citatet ovan, att studien får en hypotesprövande karaktär. Exempelvis om student A erfar ett objekt på ett visst sätt, så beror det på kontexten. Ett övergripande resultat som Berglund presenterar är att två kvalitativt olika sätt att erfar lärandet av nätverkskommunikation identifierats. Det ena är att studenterna är ”inriktade mot att uppfylla de formella kraven i en situation” det andra är ”att lära sig datavetenskap” (Berglund, 2005, s. 220).

Som tidigare nämnts liknar föreliggande studie den som Booth (1992) genomförde. Skillnaden ligger i första hand i forskningsobjekten. Booth studerade en specifik del i programmering – uppfattningar av rekursion – medan jag i denna studie studerar uppfattningar av programmeringshandlingen utan något specifikt programmeringsmoment i fokus. Till detta ska även läggas att jag valt att analysera och redovisa resultatet i ljuset av begreppet intentionalitet. Studierna skiljer sig även åt i urvalet eftersom universitetsstudenter ingår i Booths studie medan grundskoleelever från skolår 8 och 9 är föremål i denna studie. Med Berglunds (2005) studie finns den gemensamma nämnaren i att hans undersökning likt denna gör en kvalitativ ansats på objekt som är förknippade med den tekniska disciplinen. I Berglunds fall är det lärande av ett komplext datorsystem som är i fokus och i föreliggande studie programmering.

3 OM PROGRAMMERING

Alla sammanhang där en dator används har en sak gemensam, det behövs program som ger datorn instruktioner om vad den ska utföra. Papert (2003) beskriver ömsesidigheten mellan dator och program med hjälp av den Trojanska hästen som metafor. Han menar att datorteknologi kan liknas med invasionen av Troja som genomfördes med hjälp av trähästen som soldaterna gömde sig i. Det var inte hästen som var effektiv, utan soldaterna som var inne i hästen. Likväl var hästen en förutsättning för att soldaterna skulle komma in i Troja och få möjlighet att agera effektivt. Analogt ger datorn programmen möjlighet att vara ett effektivt stöd åt användaren.

I detta kapitel presenteras huvuddragen inom området programmering, det vill säga, att skapa programvara med hjälp av ett programspråk. Särskilt fokus är riktat mot vad som karakteriserar programmering med hjälp av grafiska komponenter eftersom det är den programmeringsformen som eleverna i studien använder.

PROGRAMMERING – EN ÖVERSIKT

Programmeringsbegreppet är för de flesta förknippat med digitala datorer. Det som för ett sekel sedan benämndes programmering – justeringar av mekaniska maskiner med hjälp av hålkort eller att ändra strömstyrka i elektroniska analysmaskiner – räknas idag som en allmän teknisk process (Booth, 1992). Utvecklingen av användargränssnitten har varit en bidragande orsak till att dagens datorer är betydligt enklare att programmera, jämfört med dem som vi använde för bara ett par decennier sedan. Exempelvis erbjuds idag programutvecklingsmiljöer som bygger på att användaren, med hjälp av några enkla knapptryckningar, kan skapa program. Tillgängligheten till teknik och den utbredda datoranvändningen ställer också nya krav på användaren, där behovet av kunskaper som kan liknas med programmering har växt i takt med den utveckling som gjort den digitala datorn till var mans eller kvinnas redskap. Tidigare var programmering något som enbart utfördes av experter medan denna uppgift i allt större utsträckning idag vilar på användaren. Detta gäller för såväl en privat användare som en professionell användare. För en privat

användare erbjuder de flesta program en anpassningsmöjlighet till specifika behov, en flexibilitet som mer eller mindre innebär att datorn ska programmeras för att programmen ska vara individanpassade. I detta fall talar vi inte om programmering i traditionell form utan att användaren ställer in ett program så att det passar för just honom eller henne genom att följa en guide, där olika alternativa inställningsmöjligheter erbjuds. Inom industrin ligger de flesta programmeringsmomenten idag nära produktionen. Här har robotarnas intåg gjort att programmering effektivast sker i direkt anslutning till den uppgift som ska utföras, vilket innebär att betydligt fler arbetar med programmering i sitt dagliga arbete än när programmeringen var förlagt till några fåtal personer som hade yrkesepitetet programmerare.

Samspelet mellan dator och program förutsätter att alla instruktioner som datorn tar emot och bearbetar matas in som maskinkod eller maskinspråk, som det också heter. Maskinspråket är baserat på endast två språkelement *ström på* och *ström av* (1 och 0) och det är med hjälp av dessa symboler som förutsättningarna skapas för att kunna ta emot och bearbeta data.¹⁶ På grund av sin knappa semantik är maskinspråket svårt att överblicka, förstå och tidskrävande att programmera i. Den språkklyfta som uppstår mellan människan och datorn överbryggs med hjälp av ett programmeringsspråk som ligger närmare människans språkliga förståelse än maskinkoden. Detta är också programspråkens övergripande syfte; att vara länken mellan människa och dator för att på ett så enkelt och effektivt sätt som möjligt implementera och manipulera data. Förenklingen är främst baserad på att instruktioner kan skrivas med hjälp av syntax, vilket är en blandning mellan skriftengelska och matematiska tecken i varierad grad beroende på programspråk. Programmeraren har med andra ord hjälp av ett skriftspråk istället för att programmera i maskinkod.

Booth (1992) liknar utvecklingen av programspråken som en bro mellan användaren och datorn. I ena änden av bron finns en dator och i den andra änden en människa. Ju närmre människan vi kommer desto högre är nivån på språket, för att använda terminologin inom datalogi. I programmeringssammanhang är det språken som kan ordnas under kategorin högnivåspråk som ligger närmast människans skriftspråk. Högnivåspråken är resultatet av en kumulativ programspråksutveckling som har styrts av användarnas behov av programspråk som är både lätta att lära och effektiva att använda. Det är ett

¹⁶ Begreppet *data* refererar till tecken som är bärare av information. Datorn har inte förmåga att tolka eller ge mening åt data utan endast bearbeta dem och producera ny data. Genom mänsklig aktivitet tolkas data och blir till information (Lanfegors, 1995).

stort antal programspråk som kan ordnas under familjen av högnivåspråk och vi kan även se att familjen innehåller variationer av programmeringsformer. En har redan nämnts, programmering med hjälp av skriven syntax. Till denna programmeringsform hör bland annat språken C och JAVA. Andra sätt att programmera och få in instruktionerna i datorn är med hjälp av rösten eller rörelser (se exempelvis Nordin & Wilde, 2003; Shneiderman & Plaisant, 2005), eller att använda ikoner som sätts samman till ett program. Det senare är fokus i denna studie och jag har valt att använda benämningen grafisk programmering som en beskrivning för denna programmeringsform. Andra benämningar som förekommer är Visuell programmering (Preece, 1994) eller Direkt manipulerad programmering (Shneiderman & Plaisant, 2005).

GRUNDERNA I PROGRAMMERING

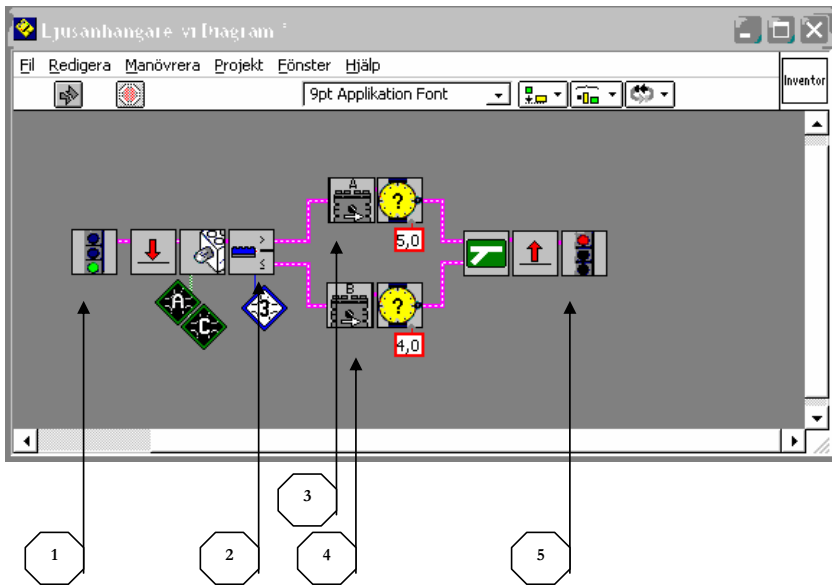
I likhet med det mänskliga språket har ett programspråk grammatiska regler som utgör strukturen i språket. Skillnaden är att ett mänskligt språk kan förstås även om det är uppbyggt av dålig grammatik, medan ett programspråk måste följa exakt grammatik för att en dator ska kunna ta emot, bearbeta och presentera data. Inom familjen av programmeringsspråk varierar de grammatiska reglerna på samma sätt som mellan exempelvis franska och svenska. Varje programmeringsspråk har sin egen syntaktiska uppbyggnad med hjälp regler kring hur ett program får skrivas och vilka betydelser specifika tecken har. Det som förenar programmeringen, oavsett språk, är att alla programmeringsuppgifter kan lösas med hjälp av enbart tre grundoperationer. Dessa grundoperationer är *sekvens*, *selektion* och *iteration*. Abelli menar att ”i princip alla moderna programmeringsspråk har konstruktioner för dessa tre grundoperationer” (2004, s. 52), vilket innebär att en lösning kan göras med hjälp av dessa operationer oberoende av vilket språk vi väljer att använda för implementering.

En *sekvens* innebär att en enskild programsats utförs genom ett antal kommandon som läses ett efter ett (steg för steg) ända till en ny instruktion ges som ändrar denna rutin. Detta kan liknas med kapitlen i en bok. Varje kapitel har sin uppgift och raderna läses en efter en ända tills en ny rutin uppstår, exempelvis att titta på en bild som illustrerar något i texten. Raderna i denna text läser du rad för rad. En ändring av rutinen *att läsa detta kapitel* görs om det mitt i ett stycke står skrivet: Fortsätt läsa i kapitel 5.¹⁷ Operationen *selektion*

¹⁷ Det ändrade typsnittet är till för att markera den text i exemplet som utgör analogin med programkod.

fungerar som en rutinändring i en sekvens där ett val mellan två eller fler värden avgör fortsättningen i programmet beroende på villkoren. Med vårt exempel kan en selektion förklaras genom att det i texten står: Läser du detta på en torsdag ska du fortsätta läsa i kapitel 5, annars fortsätter du läsa i detta kapitel. Selektion innebär med andra ord antingen – eller. Antingen är det torsdag eller så är det inte det. Den sistnämnda grundoperationen *iteration* innebär att en programsats upprepas ända tills ett villkor blir uppfyllt eller att det bryts. Åter till vårt exempel: Så länge det är måndag ska du fortsätta att läsa i detta kapitel. Villkoret som ska bli uppfyllt i detta fall för att bryta iterationen är att måndag ska övergå till tisdag om vi ser det kronologiskt.

Med hjälp av grundoperatorerna kan vi göra nästlade uppsättningar av instruktioner för att öka funktionaliteten i ett program. Med nästlad menas att en operation skrivs i en annan operation. Som ett exempel på detta kan inledningsraden på kapitel 5 lyda: Om du redan läst detta kapitel, fortsätt att läsa kapitel 6. När vi bläddrat fram till kapitel 6 står på första raden i andra stycket: Läs detta stycke, därefter fortsätter du där du var i kapitel 3. Tillsammans utgör sekvens, selektion och iteration byggstenarna för att bygga upp en programstruktur, en algoritm. I figuren på nästa sida (figur 2) visas hur dessa grundoperationer används i den programmeringsmiljö som eleverna i studien använt.



Figur 2. Exempel på hur grundoperationerna sekvens, selektion och iteration används i studiens programmeringsmiljö.

1. Programmet startar vid denna ikon och läses sedan sekvensvis åt höger.
2. Selektionen styrs i detta fall av en ljussensor. Om ljusstyrkan (luxtalet) är större än ett visst värde (som anges av programmeraren) styrs sekvensen till den övre slingan i programmet, är ljusstyrkan lägre eller lika med värdet styrs sekvensen till den nedre slingan.
3. Om ljusvärdet är större än det givna värdet som ljussensorn programmerades med, startar motor A och går i 5 sekunder. Tidsstyrningen är en iteration. Motorn ska gå så länge som tiden inte överstiger 5 sekunder.
4. Om ljusvärdet är mindre eller lika med det programmerade värdet, startar motor B med en iteration om 4 sekunder.
5. Programmet avslutas vid denna ikon.

De olika ljusstyrkorna som beskriv ovan kan exempelvis uppstå om en robot inleder sitt uppdrag genom att åka på ett grått underlag, som i detta sammanhang betraktas som neutralt underlag. Kommer sedan roboten på ett vitt underlag startar motor A enligt vårt exempel och om roboten kommer på ett svart underlag startar motor B.

PROGRAMMERING – EN FORM AV PROBLEMLÖSNING

Programmering handlar om problemlösning¹⁸ där verktyget utgörs av datorn och den programvara som skapas. För att lyckas gäller det att programmeraren under problemlösningssprocessen har förståelse för *vad* som ska utföras och *hur* detta uppnås. Vad och hur implicerar även att syftet är klargjort och att det finns en grundförståelse för vad som är möjligt att utföra med hjälp av det programmeringsspråk som finns tillgängligt. Detta kan beskrivas med hjälp av den verktygsmetaforik som Göranson m.fl. (1983) använde för att beskriva karaktären hos en dator. Använder vi oss av verktyget skiftnyckel som metafor (allmänt talar vi om datorn som ett verktyg) för att beskriva programmering skulle vi få följande beskrivning: En skiftnyckel är konstruerad för ett visst ändamål. För att kunna använda skiftnyckeln måste vi veta vad som ska utföras och hur en skiftnyckel används. Först måste man förstå i vilket syfte verktyget ska användas. Sedan måste man förstå vilket verktygets inbyggda funktionalitet är i förhållande till syftet. Därefter måste verktyget ställas in (programmeras) eller styras mot syftet enligt den funktionalitet som är möjlig. Den inbyggda funktionaliteten kan vara större hos vissa verktyg och därmed också variationen av inställningsmöjligheter och det kan också finnas fler syften med samma verktyg. Analogt kan vi med en dators funktionalitet ställa in vad som ska utföras med hjälp av datorn.

Användningen av begreppet problemlösning förutsätter att det finns ett problem (en uppgift) som ska lösas, vilket ger anledning till att definiera termen problem. Ett *problem* utgör skillnaden mellan två tillstånd, ett önskat tillstånd och ett upplevt tillstånd. Goldkuhl och Röstlinger (1988, s.16) uttrycker deras definition av problem på följande sätt:

Ett problem är skillnaden mellan ”hur jag vill att det ska vara” och ”hur jag uppfattar att det är”

Det önskade tillståndet eller ”hur jag vill att det ska vara” bestäms av situationens karaktär och problemlösarens – programmerarens – erfarenheter. Exempelvis är det svårt att programmera instruktioner för en robot som ska utföra något specifikt (förflytta sig till en koordinat) om den som konstruerar programmet inte har erfarenheter av det som ska utföras (kunskaper om ett koordinatsystem). Det är emellertid viktigt att påpeka att det inte är en

¹⁸ Problemlösning är vedertaget begrepp, därför används det i denna text. Alternativt skulle uppgiftslösning eller kunskapsprocess lika gärna kunna användas.

definierad rätt lösning i rationalistisk mening som avses i denna beskrivning av problemlösning, utan en process där en elev (programmerare) arbetar med utgångspunkt i sin egen förståelse av uppgiften.

Problemlösning ska förstås som hela den process som krävs för att en uppgift ska bli utförd, från individuell förståelse av uppgiftens art, via planering och implementering av program, till testning och utvärdering av resultatet. Detta betyder att begreppet programmering är att jämställa med begreppet problemlösningssprocess och att implementeringen av kod är en del i denna process.

En vanligt förekommande¹⁹ problemlösningssmodell inom området är den som Pólya presenterade för första gången 1945 i boken *How to Solve It - A New Aspect of Mathematical Method*. Likheter i den strukturella uppbyggnaden mellan matematik och programmering samt modellens enkelhet och generaliserbarhet har bidragit till att den flitigt används som utgångspunkt inom programmeringsundervisning. Modellen är indelad i fyra moment, där varje steg ska övervägas och analyseras i tur och ordning. Nedan presenteras en sammanfattning av Pólya (1990) modell:

- First: You have to understand the problem. What is the unknown? What are the data?
- Second: Devising a plan. Have you seen the same problem before?
- Third: Carrying out the plan (of the solution).
- Fourth: Looking back. Can you use the result?

Booth (1992) och Marton och Booth (2000) har presenterat en problemlösningssprocess som liknar Pólyas. Marton och Booth redogjorde för de olika faser som studenter fokuserar på när de ska tänka ut ett program. Redogörelsen liknar problemlösningsteori och presenteras som programmeringens skrivprocess. Processen består av tre faser; problem, specifikation och program. Problem, beskrivs som den situation som programmeraren ställs inför att lösa. För att göra detta krävs att problemet studeras och att programmeraren lär känna dess struktur så att det blir möjligt att skriva ett program. En specifikation, är resultatet av arbetet med problemfasen och är en formell

¹⁹ Att Pólyas teori är vanligt förekommande grundar jag på en genomgång av undervisningslitteratur inom programmering som jag gjort i samband med denna studie. Totalt har jag granskat 28 böcker skrivna på engelska, amerikanska eller svenska. Något fler än hälften (18) tar sin utgångspunkt i en beskrivning som är helt eller delvis identiskt med Pólyas problemlösningsteori. Några exempel på litteratur som beskriver programmering med utgångspunkt i denna teori är: Backhouse (2003), Eklund och Fernlund (1998) och Hanly och Koffman (1999).

sammanfattning av vad programmet kräver. Den sista fasen, program, är momentet då programmet skrivs. Detta görs efter att man har sett till att specifikationens alla aspekter är genomförbara i den aktuella datorn (Marton & Booth, 2000, s. 51).

PROBLEMLÖSNING OCH PROGRAMMERING – NÅGRA KRITISKA ASPEKTER

År 1637 presenterade Descartes en modell för problemlösning i sitt verk *Avhandling om metoden*. Modellen utgjorde resultatet av sökandet efter en metod som skulle vara giltig då det gällde att nå kunskap: "[...] jag försatte mig åt att söka efter den sanna metoden att nå all den kunskap som min ande var i stånd att erhålla" (Descartes, 1983, s. 48). Descartes likställde i och med sin framställan kunskapande och problemlösning. Detta är ett godtagbart likhetstecken om vi förstår problemlösning som en konstituerande process där en kvalitativ förändring av en uppfattning nås.

Descartes modell innehåller fyra regler²⁰ (steg) för att på bästa sätt lösa ett problem. I första hand är det regel nummer två som satt störst avtryck i den moderna tidens diskussioner kring problemlösning. Han hävdade att "uppdelat vart och ett av de problem jag skulle undersöka i så många delar som möjligt och som det behövdes för att bättre lösa dem" (Descartes, 1983, s. 49). Med utgångspunkt i detta måste åtminstone ett frågetecken resas. Descartes förutsätter att en uppdelning av problemet underlättar processen och som ett resultat av detta blir lösningen bättre, vilket kan förstås som att delarna var för sig kan lösas för att sedan sättas ihop till en helhetslösning bestående av ett antal separata lösningar. Resonemanget visar en avsaknad av förståelsen att helheten kan vara större än summan av delarna. Descartes utgångspunkt är ett systematiskt arbetssätt – i sin ytterligaste form – där syftet är att underlätta problemlösningen (eller att finna kunskap) men som fragmenterar innehållet till osammanhängande delar. Helhetens betydelse går förlorad enligt detta sätt att resonera.

²⁰ Presentationen av reglerna är Konrad Marc-Wogaus översättning från 1983 (Descartes, 1983, s. 49): Den första regeln är att aldrig godta något som sant, om jag inte klart insåg att det var det, eller – med andra ord – att omsorgsfullt undvika överilning och förutfattad mening [...]. Den andra är att uppdelat vart och ett av de problem jag skulle undersöka i så många delar som möjligt och som det behövdes för att bättre lösa dem. Den tredje regeln är att ge en sådan ordning åt mina tankar, att jag började med de enklaste och lättfattligaste tingen för att så småningom liksom gradvis höja mig till insikten i de mest sammansatta, och därvid förutsatte en ordning till och med mellan de ting som av naturen inte följer på varandra. Och den fjärde och sista regeln är att överallt göra så fullständiga uppräknningar och så allmänna översikter, att jag kunde vara säker på att inte ha utlämnat något.

Likväl är det just denna fragmentariska uppdelning av ett problem som många gånger tillämpas i undervisningssammanhang inom programmering vilket medför att eleverna har svårt att se hur delarna ingår i en helhet. Exempel på detta ser vi i föreliggande resultatkapitel. I undervisningen framträder det fragmentariska synsättet genom att fokus många gånger ligger på att lära sig syntaxen i ett programspråk istället för att lära sig programmering som problemlösningsmetod. En bidragande orsak kan vara att fokus har kommit att hamna på att följa modeller, så som Pólyas, steg för steg där ett moment separeras från ett annat. Jag återkommer lite längre fram till denna diskussion.

Som avslutning i detta avsnitt vill jag nämna att jag inte obemärkt kan passera diskussionen ovan utan att reflektera över kontrasterna som återfinns i detta arbete. Å ena sida framhäver jag livsvärldsfenomenologin och dess erfarenhetsfilosofiska ansats, med det erfaraende subjektet i centrum, och en förståelse av världen som reducerbar och komplex. Å andra sidan styckas denna värld – likt Descartes – upp i separata lösbara enheter i och med att problemlösningsmodeller ges en central position i teoriläggningen av programmering. En modell är en reducering av den komplexa verklighet vi lever i, en reducering som endast kan göra anspråk på att avgränsa vissa delar i ett sammanhang. Det är också svårt – om inte omöjligt - att veta var ena delen i problemlösningsprocessen slutar och var den andra tar vid. Det är tvärtom troligt att de tre första stegen i Pólyas modell sker med en samtidighet under problemlösningen där ena momentet framträder mer än de andra under ett visst ögonblick. Det vill säga att arbetsprocessen är sammanflätad, där det överhuvudtaget inte går att skilja det ena steget från det andra eftersom helheten inte kan reduceras till enskilda enheter. Avsikten med att lyfta fram problemlösningsprocessen är dels att det är ett vanligt sätt att undervisa om programmering och dels för att modellen visar var i processen som föreliggande studie har sin utgångspunkt – i implementeringsmomentet. Jag gör inte anspråk på annat än att studera aspekter av den komplexa verkligheten.

PROGRAMMERING MED GRAFISKA KOMPONENTER

Grafisk programmering innebär att den skrivna syntaxen uttrycks med hjälp av grafiska komponenter – ikoner – som sätts samman till ett sekventiellt flöde. Varje ikon representerar en funktion i programmet och de sätts samman i den ordning som programmet ska exekvera.²¹ Begreppet ikon ska i detta arbete förstås som ett tecken (bild) eller en kombination av tecken som är sammansatta så att de delar objektets kännetecken (Pettersson, 2003). Ett exempel är en ikon som föreställer en motor, den syftar till att representera objektet motor och dess funktion på roboten.

Utvecklingen av grafisk programmering kan i huvudsak kopplas till två parallella spår. Det ena utvecklingsspåret är den som skett inom industrin från behoven av enkel programmering av robotar. Parent och Lurgeau (1984) pekar på nödvändigheten av att undvika programmering som bygger på skriven syntax för att kunna effektivisera och förenkla programmeringsmomenten. Meningen är att programmeraren har en tydlig visuell överblick över den applikation som representerar lösningsmöjligheterna och genom denna överblick underlättas momentet och syntaxfelen minimeras. En annan fördel är att applikationen kan användas som en visuell prototyp innan programmet implementeras. Shneiderman och Plaisant (2005) menar att ett bra fungerande gränssnitt är en godtagbar modell av verkligheten/uppgiften, men – bör tilläggas – gränssnittet är då anpassat till att representera en viss verklighet i ett visst sammanhang.

Det andra utvecklingsspåret när det gäller grafisk programmering bygger även den på förenkling. Men i detta fall är det vardagsanvändarens behov av enkel implementering av programkod som står i centrum. Under mitten av 1990-talet har vi sett ett allt större intresse av program där användaren via ett grafiskt gränssnitt skapar programkod. Detta kan vi bland annat se genom att utveckling av webbsidor sker med hjälp av grafisk programmering där några knapptryckningar kan skapa kod som är anpassad för webben. Andra exempel ser vi inom skolan där det grafiska programmeringsverktyget är vanligt förekommande, eftersom program kan skapas utan att eleven behöver lära sig specifika programspråk som är baserade på skriven kod. Det kanske mest påtagliga exemplet är trots allt användningen av operativsystemet Windows

²¹ Med exekvering menas att ett program utför de operationer som ingår i ett program. Allmänt talar man om att programmet körs.

med tillhörande applikationsprogram. Här handlar det inte om programmering i den mening som lyfts fram i denna text, utan av användning av programvaror.

I nämnda exempel ovan finns en gemensam nämnare; tanken om att manipulation av data ska ske med hjälp av någon grafisk representation av något objekt. Vi kan samla dessa under begreppet direkt manipulation (Direct manipulation) (Shneiderman, 1986).

DIREKT MANIPULATION

Gränssnitt som bygger på direkt manipulering syftar till att synliggöra det objekt som representerar det fenomen som ska hanteras så att användaren genom ett enkelt handhavande kan åstadkomma önskad händelse. Shneiderman och Plaisant (2005, s.71) uttrycker det på följande sätt:

When a clever designer can create a visual representation of the world of action, the users' task can be greatly simplified, because direct manipulation of familiar objects is possible. [...] By pointing at visual representations of objects and actions, users can carry out tasks rapidly and can observe the results immediately (for example, dragging and dropping an icon into a trash can).

Vidare menar Shneiderman och Plaisant att programmering med hjälp av fysiska representationer (ikoner) är naturligt och efter att ha använt direkt manipulering som programmeringsform är det svårt att föreställa sig varför någon skulle vilja använda sig av en komplex syntax för att åskådliggöra en visuell process. Som en replik till deras uttalande är det viktigt att i detta sammanhang påpeka att olika programmeringsformer har olika syften och användningsområden. Grafisk programmering är inte heller utvecklad i den utsträckning så att det går att applicera på alla områden.

Användning av ikoner kan också innebära att användaren tolkar symbolen fel. En ikon kan uppfattas betyda en specifik sak, men betydelsen är något annat. Grafisk programmering innebär med andra ord ett visst mått av inlärning för att förstå vad vissa eller alla ikoner betyder. Ingen konstruktör av system, som bygger på direkt manipulation, kan begära att en bild eller ikon blir uppfattad på samma sätt oavsett användare, det vill säga, de kan inte göra anspråk på hög iconicitet.²² Att välja rätt symbolobjekt kan vara problematiskt. Till synes enkla representationer eller metaforer kan medföra att användaren blandar ihop funktioner, detta gäller speciellt om betydelsen av funktionerna ligger nära

²² Iconicitet: graden av likhet mellan en avbild och det avbildade (Pettersson, 2003)

varandra (Shneiderman & Plaisant, 2005). Griffin & Gibbs (1993) lät ett antal försökspersoner beskriva hur de uppfattade 48 olika datorikoner. Resultatet visade på att det var en tydlig skillnad mellan hur försökspersonerna uppfattade ikonerna och vad som var avsedd betydelse med samma resultat.

Fördelarna med att använda ikoner överväger emellertid nackdelarna när det gäller elever som ska lära sig programmering. En skriven instruktion (syntax) riskerar att bli fel varje gång den skrivs, på samma sätt som ett ord. Medan en ikon, som visserligen kräver visst mått av inläring av vad den representerar, aldrig kan generera syntaktiska fel.

LÄRANDE GENOM PROGRAMMERING

Denna studie handlar om ett specifikt lärande genom applikationsområdet programmering. I detta avsnitt förs en diskussion kring lärande inom detta område, det vill säga, en diskussion om vilka erfarenheter vi kan göra anspråk på att eleverna gör när de arbetar med programmering. Erfarenheter som inte är knutna till programmeringsfärdigheter, utan kunskap som är applicerbara på andra områden. I nästa kapitel återkommer jag till begreppen *kunskap* och *lärande* där min syn på kunskap och lärande i ett allmänt perspektiv presenteras.

Under de tre senaste decennierna har teorierna om programmering som medel för att lära elever hantera datorer och andra färdigheter i undervisningen varierat. Meningarna om programmering i undervisningen har sina ytterligheter från en övertro på programmeringens betydelse, till att programmeringen helt ska förkastas, eftersom det inte tillför eleverna någon kunskap förutom möjligtvis att lära sig ett programspråk (Lilja & Lindström, 2002). I början av 1980-talet var Seymour Paperts teorier om datorn som det goda lärandets verktyg både banbrytande och uppseendeväckande. Paperts vision var att eleverna skulle få kontroll över datorn genom att lära sig programmering:

I min framtidsbild programmerar barnet datorn och får i och med det både en känsla av att behärska ett stycke av den mest moderna och kraftfulla teknologi och en intim kontakt med några av de mest djupgående idéerna inom naturvetenskapen, matematiken, konsten att bygga intellektuella modeller. (Papert, 1980, s.14).

Det finns en poäng i Paperts framställning, även om vi måste förhålla oss något mer nyanserat till saken än vad som står skrivet i citatet och jag ska kommentera två saker i denna vision. För det första ställer jag mig bakom att

programmeringskunskaper kan ge en djupare förståelse för datorns struktur och funktion och genom detta kan eleverna skapa en förtrogenhet med datorn. Det är emellertid viktigt att poängtera att programmering är ett – av flera – sätt att lära sig att hantera och skapa förtrogenhet med datorn. I samband med mina empiriska undersökningar i skolan då eleverna arbetade med programmerbart konstruktionsmaterial fanns det flera exempel på elever som menade att det primära lärandeobjektet var just att de lärde sig handhavandet av datorn generellt (inte programmering). Detta är också något som visats i tidigare undersökningar med samma programmerbara material (Lindh m fl, 2003). För det andra är det knappast rimligt att djupgående idéer inom naturvetenskap, matematik och intellektuella modeller fås genom att behärska teknologin via programmering. Däremot kan kunskap om programmering och datorer bidra till att belysa naturvetenskapliga områden ur ett annat perspektiv än traditionell undervisning. Programmering ska främst användas som pedagogiskt medel i sammanhang där eleverna har nytta av programmering, om det i dessa lärandeprocesser visar sig att eleverna lär sig andra saker, får vi se detta som ett mervärde i just den specifika situationen.

ÖVERFÖRBARA KUNSKAPER TILL ANDRA OMRÅDEN

Några överförbara kunskaper – transferkunskaper till andra områden – som påvisats genom tidigare forskning är att arbete med programmering i undervisningen tränar eleverna i deras problemlösningsförmåga och logiska tänkande (Booth, 1992).²³ En annan färdighet är förmågan att förstå kausala samband, något som Papert beskriver som ”debugging” (1993, s. 52), vilket praktiskt innebär en förmåga att urskilja och rätta till orsaker till uppkomna fel i olika situationer. Det kan vara i såväl programmering som i andra praktiska problemlösningssituationer.

Papert menar även att elever som lärt sig programmera använde konkreta datormodeller för att ”tänka om tänkandet och lära om inläring” (1980, s. 33) och genom denna färdighet byggde de upp sin kunskapssyn:

Exempelvis är det många barn som bromsas i sin inläring därför att de har en modell för inlärandet där man antingen ”kan” eller ”har fel”. Men när man lär sig att programmera en dator blir det nästan aldrig rätt med detsamma. (Ibid, s.33)

²³ Booth refererar till Soloway (1985) *Why kids should learn to program*. Yale University. Originalförfattarens artikel har jag inte fått tag på.

Liknande resonemang kan vi se hos diSessa (2000) som menar att ängslan för att ha rätt eller fel begränsar kunskapsinhämtningen och att en förmåga att hantera detta kan utvecklas genom erfarenheter från programmering. Till detta bör också tilläggas att i de studier som jag granskat pekar samtliga på att kreativiteten ökar hos eleverna när de lär sig programmering. En kanske inte allt för häpnadsväckande effekt eftersom programmering inte tillhör det vardagliga skolarbetet, utan utgör många gånger ett speciellt inlägg i undervisningen, något som gör programmering till en variationsfaktor.

Lindh m.fl, (2003) visar att eleverna i första hand lyfter fram *allmän datoranvändning* som erfarenheter från arbetet med programmering. Nedan redovisas ett exempel från nämnda studie där eleverna beskriver användning av datorn som ett kunskapsområde:

Intervjuaren: Jaha okej. Har du lärt dig något som du kan ha nytta av i andra ämnen?

Elev: Ja, jo... jag tror det är mest datorkurser. Man lär ju sig mycket om datorer om man ska börja på ett företag.

4. TEORETISKA UTGÅNGSPUNKTER

I föreliggande kapitel presenteras avhandlingens ontologiska och epistemologiska utgångspunkter. Kapitlet innehåller även en diskussion kring kunskap och lärande, samt några reflektioner kring begreppet handling. Avsikten är att redogöra för vad som ligger till grund för hur jag ser på forskningsobjektet och att tydliggöra hur intervjuer och insamlad data har behandlas i ljuset av teorin. I all empirisk forskning görs antaganden om det område som är föremål för undersökning. Antaganden som utgör utgångspunkt för hur forskaren förhåller sig till forskningsobjektet och som mer eller mindre påverkar hur forskningen bedrivs. Av dessa antaganden om verkligheten och kunskapen, det vill säga ontologiska och epistemologiska ställningstaganden, följer vissa metoder medan andra utesluts (Bengtsson, 1999). Avhandlingens ontologiska grundantaganden ligger inom livsvärldsfenomenologin, en erfarenhetsfilosofisk utgångspunkt som utvecklats genom fenomenologin. Även epistemologiskt tar jag min utgångspunkt i livsvärldsfenomenologin men även fenomenografiska teorier²⁴ har inspirerat till denna teoriläggning.

LIVSVÄRLDSFENOMENOLOGI

Avhandlingens vetenskapsteoretiska utgångspunkt är vad som benämns livsvärldsfenomenologi. Livsvärldsbegreppet är utvecklat genom fenomenologin och har sin uppkomst genom den moderna fenomenologins fader Edmund Husserl (Moran, 2000; Bengtsson, 1999). Livsvärldsansatsen som ontologisk utgångspunkt innebär att vi erkänner verkligheten som komplex, sammansatt och kvalitativ, där vi befinner oss i ett historiskt sammanhang som inte låter sig reduceras till summan av tingen. Livsvärlden är öppen och oavslutad. I stället för att skilja på liv och värld förbinds dessa till en sammanflätad enhet (Bengtsson, 1999). Ansatsen ställer objektivismen och subjektivismen på kant eftersom ett särskilt intresse inom livsvärldsfenomenologin är riktat mot subjektet och objektets ömsesidiga beroende. Grundläggande är att liv och värld är oupplösliga och i detta sammanhang blir en

²⁴ Se not 3.

central utgångspunkt att det som visar sig måste ha någon att visa sig för. Bengtsson (1993) för fram sin ställning mot objektivismen med utgångspunkt i livsvärldsbegreppet. Diskussionen utgår från den matematiskt och logiskt inriktade objektivismen:

Här ersätts den konkreta livsvärlden av summan av fakta, vars främsta kännetecken är att de är utbytbara mot vilka som helst andra fakta [...] $1+1=2$ oberoende av om det är en människa och en atombomb som läggs ihop – och detta är denna objektivisms stolthet. Studiet av t.ex. människor kommer således att innebära att människan reduceras till ett objekt, till skillnad från ett subjekt, vars individuella liv och egenskaper åsidosätts (Bengtsson, 1993, s. 45).

Livsvärldsfenomenologin utgår från människans erfarenhet av den konkret upplevda världen, det levda livet. Genom våra samlade erfarenheter, våra egna eller de som förmedlats till oss, har vi lärt oss att förstå och handla i verkligheten. Schütz (1999) benämner detta som ”lager av kunskaper” (s. 32). De betydelser som framträder för oss är resultatet av att vi har olika erfarenheter av att vara-i-världen.

ORIENTERING I FENOMENOLOGIN

Termen fenomenologi användes redan under mitten av 1700-talet. Då förekom begreppet i flera texter bland andra hos Lambert och Kant (Moran, 2000). Det var under denna era flera filosofiska inriktningar började intressera sig för en gemensam fråga – läran om det sanna i kontrast till det skenbara²⁵ (Ibid.). Detta gemensamma intresse medförde att grunden lades för en fenomenologisk riktning inom filosofin som sedermera etablerades av Husserl i början av 1900-talet. Begreppet fenomen²⁶ kom att framstå som centralt och var också det som gav namn åt fenomenologin. Begreppet används i betydelsen det i sig själv visande eller det som visar sig (Bengtsson, 1999; Kroksmark, 1987).

Den moderna fenomenologi är en erfarenhetsfilosofisk ansats där intresset är riktat mot det levda livet och människans erfarenhet av den konkret upplevda världen. Sedermera kom fenomenologin att ta olika riktningar och vi kan idag

²⁵ Sken. Av tyskans *Schein*. Något som i verkligheten inte är vad det tycks vara; mer bestämt något som verkar vara verkligt, men som inte visar sig vara det. Motsatsen är dels verklighet och dels fenomen. (Filosoflexikonet, 1997, 'Sken')

²⁶ Kroksmark (1987, s.226-227) har gjort en etymologisk beskrivning av begreppet fenomen. Citat: Fenomen går tillbaka på det grekiska verbet *faínesthai* som betyder att visa sig och som gett substantivet *faínenon* som betyder det sig-visande, det tydliga. Verbet är bildat ur *faíno*, som betyder: att bringa idagen, att ställa i ljuset, vars stam *fa-* betyder ungefär: det vari något kan bli uppenbart och i sig själv synligt. Vi måste alltså med begreppet fenomen förstå: Det-i-sig-själv-visande, det uppenbara.

se en variation i den fenomenologiska rörelsen.²⁷ Trots de olika förgreningarna inom rörelsen förenas dessa i det grundläggande uppropet ”gå tillbaka till sakerna själva” som Husserl formulerade i *Logiska undersökningar* 1900-1901 (Husserl, 2000/1900, § 2 andra bandet, s.20). Med *sakerna själva* menade Husserl den konkreta verkligheten så som den framträder i vår erfarenhet i hela dess variation och mångfald. Det är erfandet av sakerna som ger oss tillgången till världen och det är genom våra erfarenheter som vi i en ständigt pågående process bildar kunskap om sakerna där för-givet-taganden sätts åt sidan:

I fenomenologin gäller det att varken ta vetenskapliga teorier, sunda förnuftet eller vilka som helst åsikter för givna; det gäller i stället att göra full rättvisa åt de objekt som är föremål för undersökning – må det vara matematiska eller logiska objekt, känslor, fysiska ting kulturobjekt, sociala institutioner m.m. Detta är sakerna själva och vår tillgång till dem är erfarenheten. Det är i dem som de visar sig och det är således utifrån erfarenheten som sakerna måste klargöras. (Bengtsson, 1987a, s. 6).

Husserls idéer kom sedan att utvecklas genom en rad filosofer. Några av de mest kända är Martin Heidegger, Max van Manen, Alfred Schütz och Maurice Merleau-Ponty. Var och en har satt sin prägel på fenomenologin och livsvärlds-fenomenologins utveckling kan bland andra tillskrivas nämnda tänkare, främst Heidegger (1927). Till utvecklingen och förståelsen av livsvärldsfilosofins betydelse på svensk botten räknar jag även Jan Bengtsson som tack vare sina sammanställningar, tolkningar och översättningar av filosofiska verk bidragit till en förståelse av den historiska utvecklingen inom fenomenologin och livsvärldsansatsens betydelse.²⁸

År 1927 publicerades ett av de mest betydelsefulla verken för livsvärlds-fenomenologins utveckling. Det var Husserls elev Martin Heidegger och hans verk *Varat och tiden* (1992/1927) som innebar en radikal förändring av fenomenologin. Heidegger ifrågasatte starkt Husserls transcendentala²⁹ fenomenologi genom att framhäva, att människan alltid är en del i världen, vi kan inte ställa oss utanför den för att betrakta den, människan är i-världen-varo. Det var också Heidegger som initierade den hermeneutiska ådran inom

²⁷ I betydelsen förändring och utveckling.

²⁸ Se exempelvis: *Fenomenologiska utflykter* (1998) och *Med livsvärlden som grund* (1999).

²⁹ Transcendental fenomenologi: Husserl utarbetade en så kallad ren fenomenologi, eller en transcendental fenomenologi som han kallade det, där han gjorde en skarp distinktion mellan vad något är (essentia) och att det är (existentia). Detta krävde ett utifrånperspektiv där existensen skulle sättas inom parentes genom hans utarbetade metod Epochén, vars syfte var att skilja existens och innehåll. Det som blir kvar utanför parentesen är det erfarna föremålets fullständiga innehåll, det vill säga ett rent fenomen (Bengtsson, 1993).

fenomenologin. Genom Maurice Merleau-Ponty utvecklades fenomenologin mot en humanisering och en tydlig koppling till existensfilosofin gav fenomenologin ytterligare utbredning. Det var i första hand och genom hans huvudverk *Phenomenology of Perception* (2002/1945) som grunden lades till utbredningen.

Inspirerad av både Husserl och Heidegger påvisade Merleau-Ponty betydelsen av att vi varseblir världen genom vår levda kropp. Merleau-Ponty menar att vi når tillgång till världen genom det kroppsliga jagets i-världen-varo och att vi inte kan vara subjekt om vi inte har en kropp, en interaktion som beskrivs som en dialog mellan kropp och värld och som får sin betydelse i varseblivningen via våra erfarenheter. Detta beskrivs som till-världen-varo (Merleau-Ponty, 2002/1945). Det är genom hans teorier som vi tydligt kom att förstå subjekt och objekt som ouplösligt förbundna. Emellertid blir det i vissa sammanhang en praktisk nödvändighet att reducera den komplexitet vi lever i till greppbara aspekter i livsvärlden för att betrakta dessa i ett mindre och ibland annat sammanhang, för att sedan återställa dem i sitt rätta element. Detta gäller exempelvis när vi ska förklara teorier om en komplex och sammansatt livsvärld. För att kunna kommunicera detta krävs att vi lyfter ut vissa aspekter ur dess sammanhang och beskriver dessa var för sig för att sedan återföra dessa till sitt rätta sammanhang – till en helhet.

Om vi återgå till sammanflätningen av subjekt och objekt så är ett vanligt exempel på detta hur ena handen håller om den andra (Bild 3). Handen eller händerna får betydelse som både subjekt och objekt samtidigt.

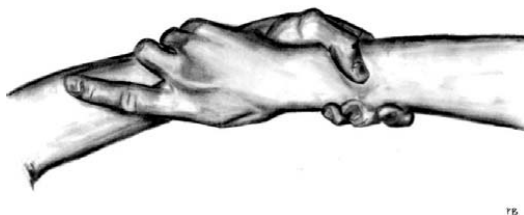


Bild 3. Illustration av Merleau-Pontys klassiska exempel på subjektet och objektets ömsesidiga beroende.³⁰

³⁰ Illustratör Patrik Bäck.

Merleau-Ponty menar att båda händerna får rollen som subjekt och objekt samtidigt och att det inte går att separera dessa och benämna den ena handen som subjekt och den andra som objekt:

[...] when I touch my right hand with my left, my right hand, as an object, has the strange property of being able to feel too. [...] When I press my two hands together, it is not a matter of two sensations felt together as one perceives two objects placed side by side, but of an ambiguous set-up in which both hands can alternate the roles of 'touching' and being 'touched'. (Merleau-Ponty, 2002/1945, p. 106).

I och med detta har Merleau-Ponty besvarat en klassisk filosofisk fråga och visat att dualismens antaganden om uppdelning mellan kroppens och tänkandets essens är felaktig, och att vi måste förstå världen som komplex och pluralistisk eftersom den överbrygger motsättningarna mellan objektivism och subjektivismen.

Med utgångspunkt i samma teoriläggning kan vi också hävda att medvetande och värld är oupplösligt sammanflätade, eftersom medvetandet alltid är riktat mot något utanför sig självt. Detta förklaras genom fenomenologins definition av *intentionalitet* som bland annat syftar till att medvetandet och världen är oupplösliga enheter (Dahlin, 2002).

BEGREPPET INTENTIONALITET

En central bestämning för såväl fenomenologin som för denna studie är *intentionalitet*. I slutet av 1800-talet införde Franz Brentano på nytt begreppet inom filosofin, en teoretisk utgångspunkt som vi finner redan under antiken hos Aristoteles (Jacquette, 2004). Begreppet kom att bli centralt för fenomenologin i och med Brentanos teoriläggning om att karaktären hos det psykiska fenomenets riktadhet pekar mot något annat än sig självt. Det psykiska fenomenet innehåller intentionalt ett objekt utanför sig självt (Haglund, 1989). Brentanos elev, Edmund Husserl, övergav så småningom de psykologiska forskningsansatserna och med det utvecklades intentionalitetsbegreppet till att vara något mer än bara en bestämning förknippat med psykiska intressen. Argumentet var bland annat att ”logiker eller matematiker arbetar inte med psykiska fakta” (Bengtsson, 1987a, s. 5).

Husserl presenterade sina teorier om medvetandets intentionalitet som en ansats till att begreppsligt fånga den sanna erfarenheten. Han ersätter det

psykiska fenomenet med att vi ska tala om ”intentionala upplevelser” (Husserl, 2002/1901, §12 tredje bandet, s.56). Alla upplevelser är intentionala och Husserl själv beskriver att dessa utgörs av en intentional akt. Denna akt är den avgränsning i vilken vi finner betydelsen av något – ett föremål, en känsla, en upplevelse etc. Husserl påpekar att det ”ibland är flera akter samtidigt närvarande och sammanvävda, men uppmärksamheten ’är verksam’ särskilt i en av dem” (Ibid, s.56). Min förståelse av detta är att resultatet av en intentional akt är den intentionala upplevelsen och att akten i sig upplever vi aldrig – det är akten som gör att vi upplever.

Fortsätter vi i Husserls fotspår kan vi med ett annat ord beskriva medvetandets riktadhet som perceptionen av något, där fenomenologins noema utgörs av vad som framträder och noesis betecknar hur innehållet framträder. Detta är vad Husserl beskriver som meningen i det som framträder, det vill säga både vad som framträder och hur detta framträder. Husserl förde fram dessa tankar som objektets två karaktärer och han menar att det inte går att göra en distinktion mellan dessa båda. Noesis är beroende av ett noema och ett noema kan inte existera utan ett noesis. Vi ser i och med detta den naturliga kopplingen till den tidigare diskussionen kring sammanflätningen av subjekt och objekt och vi kommer också att se hur detta återkommer i nästa kapitel genom fenomenografins vad -och- hur frågor. Samtidigheten hos objektet innebär att medvetandet alltid är riktat mot något annat än sig självt och att sakerna som erfars alltid erfars som något (Bengtsson, 1999). Ett exempel är då vi ser något, då är det inte seendet vi riktar oss mot, utan det vi ser. Ett annat exempel är att det inte går att tänka tänkandet. Vi kan vara medvetna om att vi har en tanke men vi kan inte tänka på tankeakten.

MEDERFARENHET

Av våra tidigare erfarenheter förstår vi att det finns något mer än det varseblivna. I samma stund som vi ser ett flygplan, presenterar sig planet som något mer än vad vi kan erfara i själva varseblivningen. Antar vi att vi betraktar planet från sidan så är det, i strikt mening, endast denna sida som presenteras för oss. Men samtidigt tar vi för givet att på andra sidan av planet finns det en ving till, vi för-givet-tar också att planet i de flesta fall har ett symmetriskt yttre så det som vi ser på denna sida finns det en spegling av på den andra sidan. Vi säger att vi *mederfar* något mer än det vi varseblir. En annan person skulle kanske mederfara att planet har bekväma säten och att det finns möjlighet att titta få film under resan eftersom denna person har erfarenhet av att denna typ

av plan är utrustat på detta vis. Grunden för skillnaderna i vad vi mederfar ligger således i våra erfarenheter. Det är vad vi mederfar som avgör hur vår uppmärksamhet är riktad, intentionaliteten framträder som det perspektiv i vilket saken visar sig.

DET INTENTIONALA OBJEKTET – INTE ETT TING

Vi talar i allmänna ordalag om vad det är som framträder och hur detta framträder för någon. När en elev programmerar uppfattas alltid innehållet som något beroende av hur eleven mederfar programmeringen. Uppfattandet (tänkandet, handlandet, kännandet etc.) är alltid riktat mot något som på olika sätt avgränsar uppfattandet av handlingen till något som utgör en central aspekt för eleven. Uppfattandet är intentionalt och blir ett subjektivt perspektiv. Det är emellertid viktigt att påpeka att ”intentionaliteten inte framträder som ett >>ting>> utan som det perspektiv i vilket tinget visar sig” (Kroksmark, 1987, s. 343) och att det inte behöver vara ett fysiskt eller verkligt objekt som vi varseblir. Exempelvis kan vi rikta uppmärksamheten mot de cykloper som Odysseus stöter på i sin irrfärd på väg hem från det Trojanska kriget till ön Ithaka. Det är knappast troligt att dessa cykloper existerar eller har existerat, men vi kan rikta vår uppmärksamhet mot dem.

INTENTIONALITETENS KATEGORISKA OBJEKT

I detta avsnitt ska jag med hjälp av ett par exempel belysa de epistemologiska antaganden som kan göras när man använder intentionalitet som ett centralt begrepp i en studie. Med andra ord, hur kan ett intentionalt objekt tänkas se ut och finns det olika kategorier av intentionala objekt? Sokolowski (2000) lyfter fram distinktionen mellan enkla och kategoriska objekt. Med enkla objekt menar han exempelvis *en hund*, medan ett kategoriskt objekt utgörs av att hunden exempelvis är *dominant*. Det senare innebär dels att hunden måste framträda på ett dominant sätt för betraktaren och dels att betraktaren har erfarenheter av skillnaden mellan dominant och inte dominant hundar. Ett intentionalt objekt – så som det ska förstås i denna studie – måste alltså utgöras av något mer än bara *en hund*, det måste inbegripa en betydelse för den som betraktar. Ett annat exempel, även det hämtat från Sokolowski (2000), är när vi betraktar en reva på en bil. Anta att uppmärksamheten riktas mot ett märke i lacken på bilen. Märket är en del i helheten och samtidigt som vi uppmärksammar revan så mederfar vi simultant att det är en bil som har revan. Revan i lacken framträder inte som en reva, utan som en reva på en bil.

Analogt är inte de objekten som framträder i elevernas beskrivningar av deras programmeringshandlingar ting eller enstaka objekt, utan de framträder i ett perspektiv i vilket sakerna visar sig.

LIVSVÄRLDSFENOMENOLOGINS BETYDELSE FÖR STUDIEN

Livsvärldsfenomenologin tar sin utgångspunkt i en kollektivt uppburen värld med ett intresse riktat mot människans erfarenheter av den konkret upplevda världen. Ontologiskt utgör ansatsen en teoretisk ram för att samla in, analysera och förstå ett empiriskt material som hämtats från elevernas beskrivningar av aspekter i deras individuellt upplevda livsvärld. Epistemologiskt kan en studie av detta slag inte göra anspråk på annat än att analysera och beskriva de fenomen som framträder genom elevernas framställningar. Kunskapsanspråket ligger i det utfallsrum av beskrivningar, som visar sig när eleverna beskriver sina uppfattningar av funktionen och uppbyggnaden i ett program.

Livsvärldsfenomenologin innebär att fenomenologin är preciserad på ett bestämt sätt. Jag som forskare är alltid en del i den livsvärld som studeras, det vill säga, att jag upplever den med mina erfarenheter, samtidigt som jag delar den med andra. Den regionala ontologin kan med andra ord beskrivas som att jag studerar aspekter av elevernas livsvärld – uppfattningar av programmering – så som den erfars på den plats där eleverna programmerar. Utgångspunkten är elevernas vardagserfarenhet av företeelsen att programmera, där livsvärlden ställer kravet på oss att något är samtidigt givet för mig som forskare och eleverna i det gemensamt upplevda sammanhanget.

Livsvärldsansatsen är metodpluralistisk i den betydelsen att det är studieobjektet som avgör val av metod och att denna används med utgångspunkt i det område som ska studeras. ”Det enda krav på metoden som ställs är att den eller de är adekvata för den verklighet som det önskas kunskap om” (Bengtsson, 1999, s. 33). Det är genom empiriska studier som vi kan få tillgång till andras uppfattningar av ett innehåll/objekt. Det vill säga, vi får tillgång till en aspekt av den levda och erfarna världen. En tillgång som gör att vi kan vidga innebörden av objektet och därmed öka förståelsen av innehållet. Med andra ord får vi ny kunskap genom att vi når en kvalitativt förändrad uppfattning av innehållet/objektet.

KUNSKAP OCH LÄRANDE

Den första fråga man bör ställa sig när man sätter en rubrik med begreppen kunskap och lärande är: Var ska man börja? En lika relevant fråga blir: Var ska man sluta? Begreppen har diskuterats i årtusenden och låter sig inte med enkelhet vare sig definieras eller överblickas historiskt. En rad diskussioner och teorier har passerat och nya kommer att dyka upp.³¹ Kunskap och lärande är några av de mest centrala frågorna i vårt samhälle, i synnerhet för alla som arbetar med utbildning i någon form. Det hävdas också att vi lever i ett kunskapssamhälle, något som Liedman (2002) och Kemp (2005) förhåller sig skeptiska till. I dagens samhälle handlar diskussionerna om kunskap i stor utsträckning om att bli expert på något område, och inte om vishet i livet. Likställer vi kunskap med någon form av expertis ”är alla människor okunniga i de flesta hänseenden och många i alla hänseenden” (Kemp, 2005, s.35). Liedman för diskussionen vidare och hävdar att kunskap är en avgörande kategori för människans liv och inte ett nytt samhällsfenomen:

Kunskap är en grundläggande kategori för mänskligt liv, inte något som enbart präglar tiden kring år 2000 då man berömmar sig av att leva i ett kunskaps-samhälle. (Liedman, 2002, s.58)

Med utgångspunkt i Liedmans och Kemps resonemang har vi alltid befunnit oss i något som kan beskrivas som ett kunskapssamhälle och att kunskap måste betraktas som något mer än att vara expert på något område. Kunskap (-en) är förutsättningen för vår överlevnad och vi behöver förhålla oss ödmjukt till begreppet och se det som komplext och mångtydigt. Behovet av att definiera kunskap kan då mer ses vara av akademisk betydelse eller som Alerby (2000, s.19) uttrycker det: ”Behovet av en definition är en definitionsfråga.” Likväl blir det i en studie som denna, där lärande med programmering studeras, nödvändigt att beskriva min syn på kunskap och lärande. Denna är grundläggande för hur studien genomförts. Min utgångspunkt tar jag i den kunskaps-syn som utvecklats inom fenomenologin och som inspirerat flertal forskare inom fenomenografin (ex. Kroksmark, 1987; Alexandersson, 1994; Marton & Booth, 2000). Med kunskap menar jag en förändrad uppfattning av något – en kvalitativ förändring. Kunskapen (och lärandet) är grundad i våra erfarenheter. Nya erfarenheter läggs till de gamla, vilket innebär att vi kan betrakta samma objekt vid olika tillfällen men aldrig med samma erfarenhet av det. Ett vanligt

³¹ En bra överblick över den pedagogiska historien och olika kunskapsnyer finns att läsa i: Kroksmark (2003) *Den tidlösa pedagogiken*.

citat av Immanuel Kants lyder: ”Att all vår kunskap börjar med erfarenheten, därom råder det inget tvivel” (Kant, 1952/1781, s.14). Det är genom vår erfarenhet som vi får tillgång till världen och därmed är det en förutsättning för att vi ska kunna erfara något.

SAMMANFLÄTADE BEGREPP

Det talas i allmänna ordalag om vilken kunskapssyn vi har och hur vi ser på lärande. Diskussionen medför många gånger en separation mellan begreppen som blir problematisk eftersom de inte kan betraktas som annat än sammanflätade. Målet med lärande är kunskap och kunskap kan inte uppstå utan någon form av lärande. Denna sammanflätning innebär att termen lärande definieras som att undervisning och inläring är samtidigt givna och att en ”lärandesituation omfattar [...] såväl den undervisande som den inlärande aktiviteten.” (Kroksmark, 2006). Däremot kan vi analytiskt separera begreppen för att å ena sidan tala om olika kunskapsformer (fakta, tyst, explicit, praktisk etc.) och å andra sidan tala om lärande i generella termer ”som en förmåga att erfara världen på ett eller annat sätt” (Marton & Booth, 2000, s. 54). Ser vi lärande på det viset innebär det dels ett sätt att ta itu med själva lärandet (hur-aspekt) och dels att det finns ett innehåll för lärandet (vad-aspekt). Detta är en utgångspunkt som i sammanhanget innebär att vad-aspekten representerar ett intentionalt objekt medan hur-aspekten utgör kvaliteten i det intentionala objektet.

Med utgångspunkt i diskussionen ovan vill jag avslutningsvis poängtera ytterligare en aspekt av begreppet kunskap. Kunskap kan inte konstrueras, det finns två grundläggande motsättningar till detta. Antingen besitter den lärande kunskapen och då behöver den inte konstrueras eller eftersökas eftersom den redan finns. Eller så saknas kunskapen. Den lärande kan i så fall inte konstruera eller söka kunskap eftersom han eller hon inte skulle känna igen den då den påträffas³² (Marton & Booth, 2000). Likväl lär vi oss saker. Svaret på denna paradox finner vi i att kunskap *konstitueras* hos oss vilket innebär:³³

³² Menons paradox. Se not 38.

³³ Jag vill föra fram några kommentarer till Marton och Booth citat som jag använder för att beskriva konstituering av något. Jag menar att konstitutionsbegreppet inte ges sin fulla rättvisa i och med att författarna beskriver detta genom ”världen runt omkring oss” och ”en del av den värld vi känner till”. Enligt mitt sätt att se det befinner vi oss *i världen*, det kan alltså inte vara något som befinner sig runt oss, analogt måste det vi lär oss vara en *del i världen*.

[...] att när vi lär oss om någonting specifikt är vi också medvetna om världen runt omkring oss. Det vi lär oss om måste bli en del av den värld vi känner till [...] den värld som redan har erfärits, är en konstituerande kraft i lärandet, precis som individens konstituerande handlingar. (Marton & Booth, 2000, s. 180)

Poängen med Martons och Booths beskrivning är att inte skilja på det inre och yttre. Det vill säga, vi kan inte se lärande som enbart något som sker inom individen (individuell konstruktion). Lika lite kan vi se lärande som något som sker enbart utanför individen (social konstruktion). Konstruktion och/eller konstruktivism i dess olika former intar därför en allt för ensidig utgångspunkt för att förklara lärande. Det är tillsammans med andra – genom dialog eller praktisk handling – som vi får distans till våra egna erfarenheter. Emellertid kräver dialogen en samtidighet mellan individuell reflektion och en kollektiv aktivitet. När vi varseblir ett objekt konstitueras alltid tingets mening i ett sammanhang och i ett tidsmässigt ögonblick. Sammanhanget och det tidsmässiga ögonblicket kan vara det samma vare sig tinget uppfattas på kollektiv eller individuell nivå, men meningen konstitueras i våra medvetanden på olika sätt beroende av våra erfarenheter.

BEGREPPET HANDLING

Studiens överordnade syfte är – som nämnts – att förstå och beskriva vad eleverna riktar sin uppmärksamhet mot under programmeringshandlingen. Utgångspunkten är med andra ord att eleverna utför en *handling*, vilket medför att begreppet blir centralt för studiens vidkommande. Genom att eleverna beskriver sin egen programmeringshandling, den som sker på datorskärmen, blir det alltså en fråga om olika uppfattningar av en handling. Handlingsbegreppet i detta sammanhang omfattar samtidigheten mellan det fysiska agerandet – placering och manipulering av ikonerna – och förståelsen för ikonernas relation och beroende av varandra.

Intention innebär en medveten och målstyrd avsikt med något. I sammanhanget svarar alltså en elevs intention mot den målinriktade och ändamålsenliga handling där valen i handlingen styrs av erfarenheter och de hjälpmedel som finns tillgängliga. Handlingen har sin förutsättning i det erfarenhetsförråd som eleven bär med sig och där tolkningen av den situation i vilken handlingen är inbegripen är avgörande för tillvägagångssättet. Alla tolkningar baseras på ett lager av erfarenheter, våra egna och de som förmedlats till oss av andra, exempelvis av lärare, syskon och föräldrar (Schütz, 1999). Det är våra erfarenheter som ligger till grund för den rationella handlingen, det vill säga en

handling som styrs av förnuftsmässighet och användbarhet och där målet är klart och distinkt. Syftet med en handling är alltid att vi ska nå framgång med det vi företar oss och det är elevernas erfarenheter av sina programmeringshandlingar som är föremål för intervjuerna i studien.

REFLEKTION I HANDLING

Med utgångspunkt i Heideggers teorier menar Winograd och Flores (1988) att programmering innebär att utforma instruktioner *tillhandsbet* så att datorn med programmet kan utföra något specifikt enligt programmerarens intentioner för ett syfte. Enligt Heidegger (1992/1927) har vi vår direkta tillgång till världen genom vår praktiska och oreflekterade erfarenhet. Vi handlar naturligt och i vissa avseenden automatiskt, detta beskrivs som *tillhandsbet*. ”Det som är till hands har över huvud taget inte blivit teoretiskt uppfattat” (Heidegger, 1992/1927, band 1, s.99), det som görs har en hänvisning till användbarheten i ett visst sammanhang. Det är först vid ett så kallat breakdown, det vill säga att vi stöter på något vi inte har lösningen på eller att något avvikande uppstår, som vi reflekterar över vårt handlande och fenomenet uppdragas *förbanden*. Ett exempel på skillnaden mellan *tillhandsbet* och *förbandsbet* ges av Heidegger genom att beskriva en person som slår i en spik med en hammare. Under själva handlingen existerar inte hammaren som don/verktyg för den som slår i spiken, den är tagen för given och finns inte primärt som ett objekt. Hammaren presenterar sig som ett objekt först när det sker någon form av breakdown, att spiken kröker sig eller att personen slår sig på tummen (Ibid).

Återgår vi till Schütz (1999) och hans teoretisering av handlingsbegreppet finner vi att begreppet *naturlig attityd* på flera punkter sammanfaller med Heideggers *tillhanden-varo*. Schütz menar att *naturlig attityd* är ett begrepp för den oreflekterade handlingen, det vill säga något som sker naturligt, utan att aktören behöver tänka på vad han eller hon gör. En framträdande tanke i Schütz resonemang är att meningen av handlandet endast kan framstå när handlandet är slutfört, så länge det pågår utförs det bara, i en så kallad *naturlig attityd*. Detta innebär att meningskonstituerandet först kan ske när man kliver ut ur den naturliga attityden. Ett resonemang som Schön (2003) skulle ha kallat för *reflection-on-action*, vilket karakteriseras av att reflektion av en handling sker efter det att handling är utförd. Det är på denna punkt som Schütz handlingsbegrepp – och Schön *reflection-on-action* – medför en viss problematik. Genom fenomenologins intentionalitetsteori är medvetandet alltid riktat mot ett meningsinnehåll i de omedelbara upplevelserna.

Vi kan alltså i en fenomenologisk utgångspunkt hävda att mening konstitueras i handling, det vill säga reflektion och lärande i handling. Denna utgångspunkt – *reflektion-in-action* som Schön (2003) kallar det – innebär att vi inte nödvändigtvis behöver reflektera, men det finns en beredskap att göra det. Schöns ståndpunkt är att vi lär och reflekterar både under och efter handlingen, men att det är olika typer av reflektioner. Reflektion under handlingen innebär att vi inte bara lär oss, vi reflekterar även över vad vi lär. Reflektion efter handlingen utgörs av vilka slutsatser som kan dras av handlingen.

5. METOD

Uppsatsens metodiska ansats ligger inom det som benämns kvalitativ forskning, där insamling av data har skett genom den kvalitativa forskningsintervjun. Ansatsen är tagen i den metodologi som är utvecklad inom fenomenografin, vars främsta intresse är att beskriva kvalitativa variationer i hur människor uppfattar något.

FENOMENOGRAFI

Under 1970-talet utvecklades en forskningstradition med utgångspunkt i antagandet att människan uppfattar företeelser på kvalitativt skilda sätt. Forskningen bedrevs av den så kallade INOM-gruppen.³⁴ Fokus riktades mot inläring och uppfattningar och bland annat studerades hur studenter uppfattar innehållet i texter (Marton m.fl, 1977). Resultatet medförde bland annat att begreppen yt- och djupinläring³⁵ fick en central betydelse för den pedagogiska forskningen. Fenomenografi fick sitt namn 1981 i och med att distinktionen mellan *första* och *andra ordningens perspektiv* introducerades som två grundläggande inriktningar inom forskningen (Marton, 1981). Marton menade att undersökningar som tar sin utgångspunkt i *första ordningens perspektiv* gör anspråk på att från observation beskriva hur något är, en objektiv beskrivning av verkligheten där fakta intar en referens för att något ska tas för sant eller falskt. Medan *andra ordningens perspektiv* tar sin utgångspunkt i människans uppfattning av ett objekt. Det blir i det senare perspektivet en beskrivningsnivå som intresserar sig för kvalitativa variationer av sakerna. Det är andra ordningens perspektiv som utgör fenomenografins mest centrala utgångspunkt (Marton 1981; Kroksmark, 1987). Skillnaden mellan perspektiven ligger med andra ord i att världen å ena sidan kan beskrivas via fakta, å andra sidan som den subjektivt erfarna världen. Detta kan åskådliggöras med hjälp av studiens forskningsobjekt. Om utgångspunkten är i första ordningens perspektiv beskriver forskaren, från de fakta som datorn är programmerad efter, observationer av sin egen uppfattning av vad som framträder som centralt för eleverna under

³⁴ INOM: Inläring och OMvärldsuppfattning. Forskningsgruppen hade sitt säte vid Göteborgs universitet och utgjordes bland annat av Ference Marton, Lennart Svensson och Roger Säljö.

³⁵ I ett uppfattningsperspektiv svarar yt- och djupinläring mot atomistiska respektive holistiska uppfattningar.

programmeringshandlingen. Andra ordningens perspektiv utgår från elevernas egna beskrivningar och resultatet blir variationen i elevernas uppfattningar av vad som framträder. Wallén (1993, s.68) menar att ”människor handlar efter sin uppfattning – inte efter hur det faktiskt förhåller sig eller hur det borde vara [...]”. Det är med andra ord från eleverna egna uppfattningar som vi kan förstå hur de begreppsbygger under programmeringshandlingen.

Ordet fenomenografi är sammansatt av de grekiska orden *fenomenon* och *grafia* och betyder i dess enklaste form att beskriva *det som visar sig för någon*.³⁶ Intresset ligger således i att beskriva hur människor uppfattar objekt i världen – objekt som visar sig som fenomen. Ansatsen gör inga anspråk på att avbilda eller beskriva någon absolut bestämning av vad ett objekt egentligen är oberoende av hur det uppfattas. Fokus ligger istället på människans erfarenhet av ett objekt. Det finns alltså en klar distinktion mellan fenomen och objekt. Fenomen är ett konstituerande mellan subjekt och objekt medan objektet är saken (Husserl, 1989). Ett grundläggande ställningstagande finner vi alltså i att vi inte kan beskriva en värld som är oberoende av våra beskrivningar eller av oss som beskriver den (Marton & Booth, 2000). Denna utgångspunkt är fenomenologisk och vi finner den hos bland andra i Merleau-Pontys resonemang när det gäller ett subjektivt varseblivande av världen. Med referens till Merleau-Ponty beskriver Bengtsson (1987b) detta och menar att världen är oupplösligt förbunden med ett subjekt och att den värld vi känner till och kan uttala oss om är ”den värld som är oss tillgänglig som varseblivande subjekt” (Ibid, s.16).

VAL AV METOD

Valet av fenomenografi har framförallt gjorts från två utgångspunkter. *Den första* är att jag är intresserad av att studera skillnaderna och likheterna i uppfattningarna av ett eller flera intentionala objekt, inte en kollektiv uppfattad essens av något. Det senare kan vi se exempel på inom fenomenologin genom det som benämns empirisk fenomenologi där ”det gemensamma består av det som förenar ett antal sätt att uppfatta och uppleva ett fenomen” (Alexandersson, 1996, s.8). Medan filosofisk fenomenologi utgår från forskaren själv och hans eller hennes

³⁶ Kroksmark (1987, s.226-227) har gjort en etymologisk beskrivning av fenomenografi. Citat: Fenomen går tillbaka på det grekiska verbet *fainesthai* som betyder att visa sig och som gett substantivet *fainemenon* som betyder det sig-visande, det tydliga. Verbet är bildat ur *aino*, som betyder: att bringa idagen, att ställa i ljuset, vars stam *fa-* betyder ungefär: det vari något kan bli uppenbart och i sig själv synligt. Vi måste alltså med begreppet fenomen förstå: Det-i-sig-själv-visande, det uppenbara. Begreppet *grafia* är också det grekiska och utvecklade ur stammen *grafi*. *Grafi* betyder *beskriva i ord eller bild, något som är, det betecknade, t ex ett stycke verklighet eller en uppfattning av denna*. I sitt förhållande till fenomen blir *grafi* en aktiv verksamhet som avbildar den studerade saken som kvalitativt skilda fenomen.

uppfattningar av ett fenomen (Ibid), det som tidigare nämnts som första ordningens perspektiv. *Den andra* utgångspunkten för metodvalet är att fenomenografin grundar sig i teoretiska utgångspunkter om kunskap och den lärande människan och dessa är viktiga för den kunskapssyn som ligger till grund för studien.

TEORETISK FÖRANKRING

I denna uppsats utgör livsvärldsfenomenologin den teoretiska grund, mot vilken fenomenografin lutar sig. Denna teoretiska underbyggnad medför att fenomenografin är beroende av fenomenologin³⁷ och inte tvärtom. Det finns flera exempel på fenomenografisk forskning som hämtat inspiration i fenomenologin några av dessa är: Jan Theman (1983) *Uppfattningar av politisk makt*, Tomas Kroksmark (1987) *Fenomenografisk didaktik* och Bo Dalin (1989) *Religionen, Själen och livets mening*.

Marton och Booth ställer frågan: Är fenomenografi fenomenologi? De svarar själva och hävdar att fenomenografin är ett ”barn i den fenomenologiska familjen” (Marton & Booth, 2000, s.154). Detta under förutsättning att ”fenomenologin definieras genom dess forskningsobjekt – mänskligt erfarenhet och medvetande” (Ibid, s.154). Både inom fenomenografi och fenomenologi är intresset riktat mot det mänskliga erfandet, där människan är given i en existerande värld. Framträdande är också gemenskapen i intentionalitetsteorin, där skillnaden ligger i att fenomenografin är en empirisk metod med utgångspunkt i andra ordningens perspektiv medan fenomenologin är filosofisk och utgår från första ordningens perspektiv (Kroksmark, 1987). Här bör det emellertid poängteras att det finns exempel på empirisk fenomenologi som tar sin utgångspunkt i uppfattningar av objekt. Exempel på hur data samlas in via intervjumetodiken inom fenomenologin finner vi hos van Manen (1990).

Syftet med studien är att beskriva det som visar sig för eleverna när de beskriver sina programmeringshandlingar. Det blir också genom detta syfte som den tydligaste kopplingen mellan fenomenologi och fenomenografi träder fram, nämligen genom fenomenbegreppet. Fenomenen visar sig i ”en kollektivt uppburen men individuellt uppfattad” (Kroksmark, 1987, s.247) livsvärld där elevernas livsvärldar utgör sammanhanget i vilket uppfattningarna träder fram.

³⁷ Jag likställer inte fenomenologi och livsvärldsfenomenologi (se kapitel 4). I denna diskussion använder jag mig av fenomenologi som utgångspunkt eftersom detta utgör den livsvärlds-fenomenologiska kontexten och att tidigare avhandlingar och arbeten som använt denna ansats ligger mer eller mindre i närheten av livsvärldsfenomenologins grundantaganden.

Studien medför även en koppling till den tekniska disciplinen genom att programmering av datorer är i fokus. Forskning inom denna vetenskapsgren ligger nästan uteslutande inom positivistiska ramar, där kausala orsaksbestämda samband som går att mäta studeras (Bjurwill, 1995). Inom denna inriktning förbises ofta människans erfarenheter av att använda tekniska artefakter, erfarenheter som är avgörande för hur kunskap konstitueras i dessa situationer. Heidegger (1974/1962) hävdar att erfandet av tekniken är nödvändig och tillhör teknikens sfär. Tekniken kan alltså inte enbart ses som instrumentell eftersom erfandet av det instrumentella ger tekniken dess betydelse. När denna studie initierades var intresset riktat mot människan och hennes erfarenheter. Det var därför viktigt att hitta en teori och en eller flera metoder som ger stöd åt denna utgångspunkt. Livsvärldsfenomenologin ger studien den teoretiska grund som behövs för att förstå det emiriska material som samlats in och analyserats med stöd i fenomenografin.

CENTRALA BEGREPP INOM FENOMENOGRAFIN

Människans uppfattning av något är det mest centrala inom fenomenografin och begreppet *uppfattning* utgör således en av ansatsens grundstenar. Uppfattningsbegreppet i fenomenografisk mening är något annat än vad vi i vardagssammanhang kallar uppfattning. Skillnaden är att en fenomenograf analyserar uppfattningar *av något* där människans grundläggande förståelse av något är i fokus. En uppfattning i allmän betydelse innebär ofta att en värdering vägs in i en medveten reflektion *om något*.

Uppfattningsbegreppet består av en *vad-* och en *hur-aspekt*. Dessa termer har även en direkt koppling till didaktiken genom att vi talar om att *vad-aspekten* visar på ett innehåll, där *hur-aspekten* visar processen som leder fram till karaktären hos det konstituerade. Då vi studerar människors uppfattningar av något avses att beskriva *vad* och *hur* något visar sig för någon. I denna betydelse innebär en uppfattning att konstituera mening och ligger på så sätt nära kunskapsbegreppet. Behandlingen av ett innehåll (*hur*) utgör en viss avgränsning av innehållet (*vad*), en avgränsning som är beroende av våra erfarenheter och som utgör ett riktat meningsinnehåll. Detta innebär att vi kan se en koppling mellan begreppen *uppfattning* och fenomenologins *intentionalitet*. Kroksmark (1987, s.232) menar att ”analytiskt består en uppfattning av en *intentional akt*. Med intentional menas att medvetandet är riktat mot något speciellt.” När människor uppfattar något behandlas samma innehåll (objekt)

men skilda kvaliteter avgränsas av innehållet och det är dessa kvalitativt skilda avgränsningar som är föremål för den fenomenografiska studien.

FRÅN ATT UPPFATTA TILL ATT ERFARA

I senare tids fenomenografiska texter talas det snarare om att *erfara* än att *uppfatta* något. Marton och Booth (2000, s.146) menar att ”den fenomenografiska forskningens grundenhet är ett sätt att erfara någonting [...] och forskningens objekt är variationen i sätt att erfara fenomenen”. Detta kan jämföras med att fenomenografin tidigare förklarades som ”läran om de kvalitativt skilda sätt på vilka människor uppfattar olika aspekter av sin omvärld” (Marton, 1986, s.67).

Att erfara något menar Marton och Booth (2000) är att *något är någonting för någon* och att meningen i det som erfars är dialektiskt sammanflätad med en struktur, det vill säga att något är relaterat till ett sammanhang. Förskjutningen från att *uppfatta* till att *erfara* kan förstås som att erfandet av något inbegriper något mer än att uppfatta något. Marton och Booth ger själva ett exempel på *att erfara* inbegriper förstå, uppfatta, begripa etc. Ett antagande är att denna svängning innebär en tätare anslutning till fenomenologins sätt att använda begreppet erfara och även en uppgörelse med metoden *fenomenografi* till förmån för en tätare teoretisk förankring genom *variationsteorin*. Det är också genom detta antagande som jag hävdar att uttrycket ”aspekter av sin omvärld” ersatts av ”dialektiskt sammanflätad” i en strävan att skriva sig från något som kan uppfattas som ett dualistiskt förhållningssätt. Begreppet omvärld som tidigare använts kan inte förstås som annat än en separation mellan människa och värld. Detta blir också tydligt i och med att Marton och Booth hävdar att ”ett erfande är till sitt väsen icke-dualistiskt” (s.160) även om termen icke-dualistisk är olycklig i sammanhanget, eftersom det mycket väl kan innebära att erfandet är exempelvis idealistiskt eller materialistiskt så länge det inte är dualistiskt.

Min teoretiska utgångspunkt är att människa och värld är ömsesidigt beroende, där världen är beroende av liv, och livet beroende av världen (Bengtsson, 1999). Förankringen i fenomenologin torde innebära att begreppet erfara användes framför uppfattningsbegreppet. I detta arbete har inget sådant ställningstagande gjorts, jag ser snarare en fördel med att använda båda begreppet eftersom de är sprungna ur olika ansatser. Begreppet *uppfatta* har varit dominerande inom fenomenografin under snart tre decennier. Det vore fel och något konstruerat

att diskutera fenomenografin enbart från begreppet *erfara*. Det är också mot teoretiseringen av uppfattningsbegreppet som forskningsfrågan vilar i denna studie. Samtidigt har de teoretiska inflödena från fenomenologin medfört att vi hellre talar om att erfar något eftersom det inbegriper något mer än det kognitivt inriktade uppfattningsbegreppet. Att erfar något enligt den fenomenologiska terminologin innebär att vi med hela vår kroppslighet har ett världsligt medvetande.

KRITISKA ASPEKTER

Varje vetenskapsteoretisk utgångspunkt, metod eller enskilda teoretiska antaganden utsätts för kritiska röster där ifrågasättandet oftast utgår från något annat teoretiskt perspektiv. Vore det inte så, skulle vi enbart ha en teori att utgå från och det skulle inte vara av betydelse att studera människors olika uppfattningar av aspekter i världen. I detta avsnitt belyses några kritiska röster mot fenomenografin. Kritik som är viktigt att lyfta fram och bemöta för att den ger näring åt den vetenskapliga diskussionen, men också för att visa att dessa uppmärksammats i metodvalet.

Viss kritik har riktats mot fenomenografin gällande att resultatredovisningen lätt kan upplevas som positivistisk. Englund och Svingby (1986) menar att kategoriseringen av elevers uppfattningar av ett innehåll riskerar att konservera ämnesuppfattningen och att dessa kan förstås som givna sanningar om inte bakomliggande orsaker tydliggörs. Detta påstående innehåller två delar. Dels att kategoriseringen tas för givna fakta och dels att kategorierna inte tar hänsyn till bakomliggande faktorer. Det första, missförståndet att kategorierna ses som givna fakta, ligger i första hand hos betraktaren och dennes tolkning av resultatet. Istället för att lyfta fram enskilda uppfattningar inom ett utfallsrum bör betraktaren förhålla sig till hela utfallsrummet som resultat (Dahlin, 1989). Den andra delen i kritiken, avsaknaden av bakomliggande faktorer, placerar sig i diskussionen om kausala orsakssamband. I studier där man intresserar sig för bakomliggande orsaker – exempelvis inom psykologi eller fysik – fokuserar på varför ett beteende uppstått eller varför en fysisk kropp ser ut som den gör. I detta perspektiv tenderar resultaten att bortse från hur objekten visar sig för människan, det vill säga objekten beteende och fysisk kropp hamnar i skuggan av de kausala grunderna. Resultatet av en fenomenografisk undersökning ska inte ses som en studie som på kausala grunder försöker beskriva givna fakta eller strukturer hos en människas tänkande. Den ontologiska kritiken mot detta

är att ”tänkandet egentligen aldrig kan förklaras kausalt, eftersom det alltid är tänkandet som förklarar” (Dahlin, 1989, s.39).

Ytterligare kritik är att fenomenografien i vissa avseenden framstår som teorilös eftersom man kan finna fenomenografiska studier som vilar mot skilda teoretiska inflöden. Denna inkonsekvens bör emellertid ses som en förtjänst snarare än en brist:

[...]fenomenografien anpassas till förändlig verklighet och till skilda innehåll i stället för att via en snäv teoriram försöka anpassa verkligheten till en vetenskaplig modell. (Kroksmark, 1987, s.225)

Fenomenografins anspråk på anpassning till den konkreta levda verkligheten innebär att vi måste ta stöd av olika teoretiska inflöden. Det är inte heller konstigt om en metod (metodologi) vars intresse är att beskriva variationen av uppfattningar är öppen för en variation av teoretiska inflöden och att vi snarare ska tala om metoder än metod. Trots variationen finns ett gemensamt förhållningssätt som utgör grunden för all fenomenografisk forskning, ett förhållningssätt som främst rör hur fenomenet behandlas. Det är forskaren som avgränsar det objekt som är centralt för studien och samtidigt vilar det på forskarens ansvar att behandla fenomenen med en sådan öppenhet och följsamhet så att det leder till en vidare utveckling. För att fenomenen ska kunna behandlas på detta sätt krävs att vi har viss kunskap om objektet (Marton & Booth, 2000). Detta medför ett ställningstagande när det gäller frågan om forskaren bör ha kunskaper om det objekt som studeras eller inte. Enligt min mening är det viktigt att vi har kunskaper om det objekt som behandlas för att kvalitativt skilda uppfattningar ska upptäckas både under intervjun och under analysen. Vi skulle annars stå inför en situation som liknar det som benämns Menons paradox.³⁸

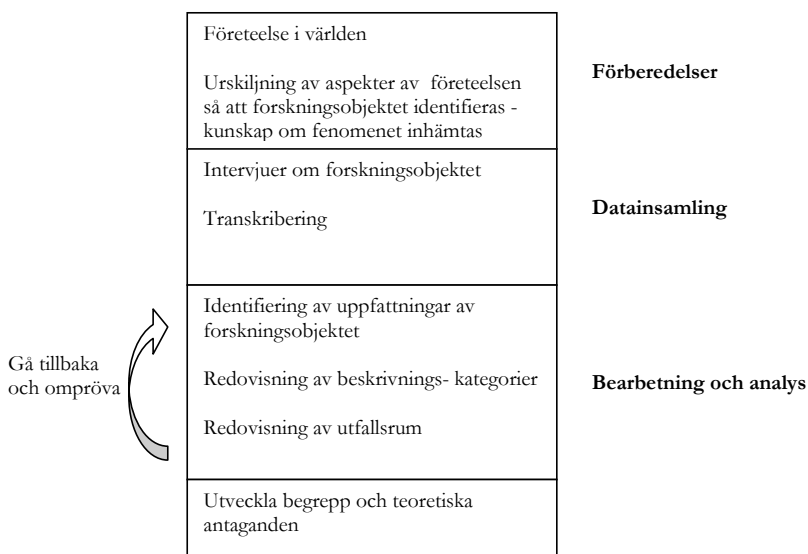
DEN FENOMENOGRAFISKA ARBETSGÅNGEN

Den metodiska arbetsgången inom fenomenografien kan variera med undersökningens syfte och innehåll, det är studieobjektets struktur som avgör den specifika arbetsgången. Några bärande huvudmoment i arbetsordningen är

³⁸ Citatet om Menons paradox är hämtat från Marton och Booth (2000, s.16). Citat: /.../ Det är i en dialog mellan Sokrates och Menon, en ung man från Thesaloniki på besök i Athen, och den börjar med att Menon ställer frågan om dygd kan läras. Sokrates svarar att han inte ens vet vad dygd är, och hävdar att Menon inte heller gör det. Sokrates föreslår att de tillsammans skall ge sig i kast med att söka efter ett svar, men Menon kommer med en invändning som blivit känd som Menons paradox: ” Hur kan man söka efter någonting när man inte vet vad det är? Man vet inte vad man ska leta efter, och om man skulle stöta på det, skulle man inte känna igen det som det man letar efter.”

METOD

emellertid likartade i nästan alla fenomenografiska undersökningar, en grov indelning är: *förberedelse*, *genomförande av datainsamling* och *behandling av data*. Några exempel på schematiska modeller över de ingående arbetsstegen finner vi hos Uljens (1989, s.11), Booth (1992, s.58) och Alexandersson (1994, s.73). Redogörelsen nedan är inspirerad av nämnda författares beskrivningar av de komponenter som det fenomenografiska forskningsarbetet består av. Beskrivningen (figuren) kan inte ses som annat än en förenkling av vad det innebär att arbeta med fenomenografi praktiskt. Den ger emellertid en bild av arbetsgången och hur de ingående komponenterna förhåller sig till varandra och helheten.



Figur 3. Schematisk beskrivning av den fenomenografiska arbetsgången.

Modellen ska läsas uppifrån och ned och har sin utgångspunkt i att en företeelse i världen avgränsas så ett forskningsobjekt kan identifieras. I nästa fas genomförs intervjuerna följt av utskrift av intervjuerna. Intervjuerna är mycket centrala inom fenomenografien där målet är att komma åt olika aspekter av en företeelse. I den sista fasen består arbetet av bearbetning och analys av intervjuerna. Arbetet omfattas av en sträva att förstå meningsinnehållet i materialet och genom att jämföra likheter och skillnader i försöker man kategorisera kvalitativt skilda grupper av uppfattningar. Analysarbetet omprövas

för att prova noggrannheten i kategorierna innan resultatet redovisas i form av beskrivningskategorier. Målet med den fenomenografiska undersökningen är att utveckla begrepp och teoretiska antaganden av den företeelse som var föremål för undersökningen.

DEN KVALITATIVA FORSKNINGSINTERVJUN

Den datainsamlingsmetod som bäst stödjer studiens syfte är den kvalitativa forskningsintervjun. Intervjuformen betecknas som halvstrukturerad, vilket innebär att diskussionsteman och förslag till uppföljningsfrågor är uppställda på förhand, men med en möjlighet till förändringar för att kunna följa upp svaren (Kvale, 1997). De enskilda frågorna som intervjun kommer att behandla är i de flesta fall inte bestämda på förhand. De växer successivt fram under intervjuens gång genom att den intervjuade definierar innehållets *vad* och *hur* inom ett temaområde. Den innehållsavgränsning som är föremål i denna studie nås genom att låta eleverna beskriva sin uppfattning av deras egen programmeringshandling.

Den halvstrukturerade intervjun präglas av öppenhet och följsamhet i frågandet och i hur svaren hanteras. Här vilar ett ansvar på den som intervjuar att avgränsa objektet – ett objekt som ständigt ändrar karaktär – under intervjuens gång så att parterna pratar om samma sak. Det är i detta skede som fenomenografien riktar ett särskilt intresse mot vad informanten definierar som innehåll (*vad*) och hur detta innehåll visar sig. För studiens vidkommande utgörs detta *vad* och hur av vad eleverna riktar sin uppmärksamhet mot i sina beskrivningar av programmeringshandlingen och hur detta framträder. Informanten styr intervjun utifrån sitt sätt att definiera dessa aspekter, men det vilar på forskaren att se till att fokus bibehålls och om så behövs styra tillbaka intervjun mot det innehåll som informanten inringat i ett tidigare skede. På så vis är en intervju alltid intentional i någon mening eftersom forskaren strävar efter att hålla fokus.

Karaktären på intervjun kan ta många riktningar och ofta innebär ett möte av detta slag en känslomässig och ibland psykisk prövning både för forskaren och för informanten. Detta kan exempelvis ta sitt uttryck i osäkerhet inför och under intervjun. Även den mentala urladdning som en blottläggning av sina tankar innebär kan medföra en tomhetskänsla efter intervjun. I de flesta fall ramas intervjun in av perioden mellan det att inspelningsutrustningen sätts på

tills det att den stängs av, men de psykiska och känslomässiga effekterna kan följa som ett eko efter intervjun (Warren, 2002).

Det vanligaste sättet att dokumentera intervjuer är med hjälp av en bandspelare. Fördelen är att forskaren kan säkerställa att allt som sagts under intervjun finns dokumenterat och att intervjun kan ske med fokus på samtalet utan att forskaren behöver anteckna eller komma ihåg det som sagts. I fenomenografiska studier är ljudupptagning viktig dels för att forskaren ska ha möjlighet till att gå tillbaka för att lyssna på samtalet i efterhand och dels för att kunna transkribera (ljud blir till text) hela intervjun som underlag för analysen. Behovet av att kunna lyssna på intervjun i efterhand är exempelvis för att kunna upptäcka nyanser i informantens uttalanden som kan vara viktiga i analysen.

BESKRIVNING AV DATAINSAMLINGEN

Innan studien påbörjades gjordes provintervjuer för att se hur data på lämpligast sätt skulle samlas in. Eftersom målet var att komma så nära den spontana handlingen som möjligt var utgångspunkten att eleverna själva skulle berätta hur de tänkte då de programmerade. Under provintervjuerna testades olika tillvägagångssätt för att fånga eleverna sätt att tänka. I ena fallet fick eleven programmera först, följt av att vi satte oss för en intervju. I detta fall var det påtagligt problematiskt att koppla vissa utsagor till specifika delar i programmet, jag var inte säker på att vi talade om samma sekvens i programmet eller om samma ikoner. Intervjun handlade mer om att reda ut vad vi talade om än att förstå hur eleven tänkte under programmeringshandlingen. För att lösa detta flyttade vi oss under intervjun till datorn där programmet fanns. Följden blev att eleven började klicka med musen och koncentrationen gled mer och mer över till programmet på datorskärmen. Som en följd av detta hörde eleven inte frågan vid ett par tillfällen. Vid den andra provintervjun lät jag eleven beskriva vad som hände under programmeringshandlingen, en så kallad tänka högt situation och för att få ett djup i intervjun var jag tvungen att följa upp elevens beskrivningar med följdfrågor. I detta fall upplevde jag att frågorna ibland störde eleven och det var påtagligt svårt för eleven att koncentrera sig på både intervjun och programmeringen. Antingen programmerade eleven eller så svarade han eller hon på frågorna.

Erfarenheterna från de två tidigare provintervjuerna resulterade i att en tredje variant (med samma elev som den första prov intervjun genomfördes med) provades. Under tiden som eleven programmerade videofilmades datorskärmen

där programmet konstruerades. På så sätt dokumenterades alla förflyttningar av ikonerna, hur dessa placerades i förhållande till varandra och om eleven stannade upp för att tänka eller byta ut någon ikon mot en annan. Filmen var sedan utgångspunkt under intervjun. Tillvägagångssättet fungerade bra och valet föll på att använda detta arbetssätt för att samla in data till studien. Emellertid gjordes några justeringar inför den första intervjun som ingick i studien jämfört med den sista provintervjun. Den viktigaste förändringen var att det ordnades en brytare som var kopplad till videokameran så att både jag och eleven hade möjlighet att stanna videobandet under intervjun. Detta gjordes främst för att eleven skulle känna att han eller hon hade kontroll över situationen och att intervjun inte skulle kännas stressande.

MÖTET MED ELEVERNA

Innan intervjun förklarade jag noggrant förutsättningarna för mötet. Ramarna för intervjun och alla moment: programmering, filmning, förflyttningen till tv, filmen och intervjun klargjordes så att inga överraskande moment skulle uppstå för eleven under mötet. Därefter informerades eleven om hur intervjun är upplagd och vad syftet med undersökningen var och att han eller hon när som helst kunde avbryta om något kändes obehagligt.

Studien tar sin utgångspunkt i situationen där jag har gett eleven ett uppdrag att programmera en robot som ska lösa en uppgift. Mötet skedde i ett rum där bara jag och eleven befann oss – det har under studien varit i ett klassrum eller ett grupprum. Mötet inleddes med en diskussion om allmänna och personliga saker så som: intressen, bakgrund och spontana saker som kom upp. När det var dags för programmeringsmomentet fick eleven en färdigbyggd robot (se bild 1, s.18) Syftet var att alla elever som ingick i studien skulle ha samma förutsättningar när det gäller robotkonstruktion. När programmeringsuppdraget presenterades fick eleven två varianter av uppgifter att välja mellan. Båda hade samma svårighetsgrad, skillnaden var den att olika komponenter på roboten krävdes för att lösa uppgiften – ljussensor eller trycksensor. Anledningen till att det fanns två olika uppgifter var att om en elev exempelvis inte hade arbetat med ljussensor fanns istället en möjlighet att lösa uppdraget genom att programmera roboten så att en trycksensor användes istället. Under programmeringen filmades applikationen (skärmen) där programmet konstruerades och intervjun skedde sedan när filmen spelades upp på en TV.

METOD

Att använda stimulus för att samla data kallas för *stimulated recall*, ett metodiskt grepp som bland andra Alexandersson (1994) använde då han undersökte vad lärare riktar uppmärksamheten (medvetandet) mot när de undervisar. Syftet med att använda videofilmen som stimulus i denna studie var att eleven under intervjun skulle få en bild – en utgångspunkt – för att kunna beskriva hur de tänkte under programmeringshandlingen. Ett annat syfte var att skapa en visuell referenspunkt som möjliggjorde att jag och informanten utgick från samma objekt under intervjun.

INTERVJUN

Under intervjuerna använde jag mig i första hand av tre frågeteman i syfte att ringa in och förstå kärnan i studieobjektet, det vill säga hur de intentionala objekten beskrivs av eleverna. Frågeområdena utgjorde ett stöd för att hjälpa till att hålla fokus under intervjuerna men det är svårt att beskriva i detalj hur respektive fråga ställdes eftersom varje intervju är unik och samtalens innehåll styrde intervjuns nästa fråga. Följande frågeteman var utgångspunkten under intervjuerna:

- Helhetsbeskrivningar. Eleverna skulle beskriva hur de tänkte under programmeringshandlingen utan att vissa specifika sekvenser lyftes fram.
- Beskrivningar med utgångspunkt i delar i programmet. Vissa sekvenser lyftes fram för att eleverna skulle beskriva hur de tänkte under dessa delar. De sekvenser som jag såg som särskilt intressanta var antingen om eleven funderade extra mycket eller om några ändringar i programmet gjordes under hand som eleven programmerade.
- Beskrivningar av vad som är viktigt att tänka på under programmeringshandlingen. Under detta frågeområde hamnar även de samtal där eleven uppmanas att beskriva för en utomstående hur man tänker när man programmerar och hur eleverna uppfattar att de lär sig programmering.

Frågeområdena har tillsammans bidragit till att ge den bild av *hur* och mot *vad* eleverna riktar sin uppmärksamhet under programmeringshandlingen.

STIMULATED RECALL SOM STÖD VID DATAINSAMLING

Theman (1983) pekade på problematiken med att intervjua människor om en händelse som låg bakåt i tiden. Han menade bland annat att det är viktigt att återkommande under intervjun hänvisa till det objekt som var föremål för diskussionen för att säkerställa att de talade om samma sak – i nämnda fall uppfattningar av politisk makt. Theman menade genom att nämna objektet återkommande får detta också en minneseffekt, att informanten lättare kan dra sig till minnes av det som hänt. Tillvägagångssättet som Theman använde kan beskrivas som en verbal *stimulated recall*.

Alexandersson hävdar att *stimulated recall* kan ses som en stödmetod för ”att påminna en person om det egna tänkandet under en tidigare episod” (1994, s.33). Metoden har framförallt används inom forskningsområdet Teacher Thinking som behandlar lärares tänkande i sin undervisningssituation. Inom det forskningsintresset är fokus riktat mot lärarnas beskrivningar av sitt sätt att undervisa på och hur de tänker i undervisningssituationen (Calderhead, 1981). Det är viktigt att påpeka att jag inte ser *stimulated recall* som en metod i samma mening som exempelvis fenomenografi som bygger på vissa gemensamma moment i arbetsgången. Begreppet bör betraktas som ett komplement till den metodiska arbetsgången för att kunna samla in data i vissa speciella situationer. Undersökningar som genomförs med hjälp av *stimulated recall* kan skilja sig åt väsentligt, bland annat beroende av intervjuteknik och grundläggande epistemologiska antaganden (Haglund, 2003). Fördelen med att använda *stimulated recall* är – som jag beskrev tidigare – att det möjliggör för att forskaren och informanten att ha en gemensam referens som utgångspunkt under intervjun för att underlätta identifieringen av vilket objekt som avses i utsagorna. Arbetssättet underlättar även när forskningsintresset är riktat mot beskrivningar av en handling, sin egen eller någon annans, eftersom informanten i efterhand har möjlighet att beskriva ett händelseförlopp som tidigare ägt rum. Likväl är det inte utan en viss reservation som jag lyfter dessa fördelar, argumenten kan även användas som bränsle för den kritik som riktats mot *stimulated recall*.

I sammanhanget är det två kritiska aspekter med att använda film som stimulus som bör lyftas fram. Den första är att filmen i sig är data, vilket innebär att eleverna försätts i en situation där de tolkar sina egna handlingar. Beskrivningsperspektivet flyttas från programmeringshandlingen till tolkning av

programmeringssituationen. Yinger (1986) menar att den videoinspelade situationen är en representation av den identiska och att de visserligen till sin karaktär är snarlika men att intervjun i första hand kommer att beskriva det stimulus som spelas upp. Den andra aspekten är att det finns en risk att elevernas utsagor speglar deras färdigheter, det vill säga deras skicklighet i programmering och inte hur de tänkte under handlingen. Dessa aspekter går inte att bortse från, de är inte heller meningsfullt att ens försöka dementera dem, det är snarare viktigt att de finns med i bilden som möjliga påverkans faktorer. Likväl menar jag att inga andra anspråk görs än att återge elevernas egna beskrivningar av deras programmeringshandling och att *stimulated recall* har varit ett sätt att komma åt forskningsobjektet.

URVAL

Undersökningsgruppen är baserat på nio elever från skolår 8 och 9 i svensk grundskola och förlagd vid tre skolor i mellansverige. I samråd med respektive lärare i klasserna har elever som arbetat med konstruktionsmaterialet under en längre sammanhängande period och där programmering har varit ett framträdande inslag för eleverna valts ut. Urvalet har med andra ord varit styrt med motivet att elever med likartade programmeringsfärdigheter var önskvärt för att jämförelsen av data skulle vara betydelsefull, med koncentration på elever som kan programmera. Motivet var att eleverna skulle ha förmåga att lösa de problem (breakdown) som eventuellt uppstod. Styrning av detta slag kan vara problematiskt där risken är att en studie tenderar att bara gälla en viss grupp av individer, vilket gör att resultatet är svårt att betrakta i ett vidare perspektiv. Enligt min mening föreligger denna fara om urvalet gjorts allt för snävt utan hänsyn tagen till hur variationen inom hela gruppen bedöms vara. I föreliggande fall anser jag att jag hade god uppfattning om programmeringskunskapernas variation hos de elever som deltog i projektet, inte enbart de som deltog i undersökningen. Jag kan därför se en fördel med att studera en grupp elever som jag på goda grunder kan anta är relevanta för undersökningens syfte. Med ett slumpmässigt urval hade risken varit stor att jag inte kunnat närma mig forskningsobjektet på samma sätt, en del intervjuer hade kanske skett med elever som inte vill delta eftersom de känner sig osäkra eller inte kan programmera.

De programmeringsfärdigheter som söktes hos eleverna som skulle delta i undersökningen kan beskrivas ligga mellan kompetent utförare (Competence) och skicklig utförare (Proficiency) (Dreyfus & Dreyfus, 1986). Dessa stadier

ingår i en modell för inläring av färdigheter där vägen går från att följa explicita regler till intuitiv inläring.³⁹ *Kompetent utförare* karaktäriseras av att eleven känner igen fler och fler element från tidigare situationer. Emellertid kan eleven känna att fakta och regler är överväldigande där känslan för betydelsen av fakta och regler saknas. Det krävs fortfarande en viss ansträngning av eleven för att utföra eller beskriva en handling. *Skicklig utförare* kännetecknas av en mer oreflekterad kunskap, som naturligt kommer till uttryck i situationen på grund av individens användning av tidigare erfarenheter. Eleven har förmåga att se holistiskt på situationen men tänker fortfarande analytiskt på vad han eller hon ska göra (ibid.). Det är viktigt att påpeka att Dreyfus och Dreyfus poäng med de olika stadierna är att oberoende nivå på kompetens utvecklar människan kunskaper och färdigheter som är bundna till deras förståelse av sitt arbete. Deras terminologi för olika kompetensnivåer används i detta fall för att beskriva en förväntad kunskapsnivå hos de elever som valdes ut. Kompetensstadierna var utgångspunkten i samtalen med lärarna som undervisade eleverna med materialet så att lärarna kände till mina önskemål om vilka elever som skulle ingå i undersökningen. Det var sedan respektive lärare som valde ut eleverna.

ETISKA ÖVERVÄGANDEN

De etiska övervägandena är noggrant beaktade och bygger på fyra huvudkrav för forskning där människor är involverade och där individskyddskravet är väsentligt. Huvudkraven som legat till grund för studiens utformning är de regler som presenteras av Vetenskapsrådet (2006). Punkterna nedan är en sammanställning av dessa regler:

- *Informationskravet.* Forskaren ska informera uppgiftslämnaren om deras uppgift i projektet och den aktuella forskningsuppgiftens syfte.
- *Samtyckeskravet.* Deltagarna i en undersökning har rätt att själva bestämma över sitt deltagande och i vissa fall bör samtycke dessutom inhämtas från vårdnadshavare (exempelvis om deltagaren är under 15 år och undersökningen är av känslig karaktär).
- *Konfidentialitetskravet.* Uppgifter om alla som ingår i en undersökning skall ges största möjliga konfidentialitet. Detta krav har ett nära samband med frågan om offentlighet och sekretess.

³⁹ Modellen bygger på fem stadier: Novice, Advanced beginner, Competence, Proficiency and Expertise. Färdigheterna hos den lärande går enligt denna model från *vetta att* till ett erfarenhetsbaserat *vetta hur*.

METOD

- *Nyttjandekravet.* Uppgifter om enskilda personer insamlade för forsningsändamål får inte användas eller utlånas för kommersiellt bruk eller andra icke-vetenskapliga syften. Uppgifterna får inte heller användas för beslut eller åtgärder som direkt påverkar den enskilde utan medgivande av den berörda.

I varje undersökning av vetenskaplig karaktär ska forskaren avväga värdet av det förväntade kunskapsstillskottet mot möjliga faktorer som kan bedömas vara risker för de inblandade och eventuellt för tredje person. Praktiskt har beaktandet av *informationskravet* inneburit, att jag noggrant informerat respektive elev, såväl innan besöket vid skolan som vid själva intervjutillfället, om projektets syfte och vad intervjun kommer att behandla. Innan intervjun förklarades vilka moment som eleven skulle stöta på och att han eller hon när som helst kunde avbryta om vederbörande på något sätt inte ville fortsätta. Då projektet startade togs det hänsyn till *samtyckeskravet* i form av ett utskick till respektive elevs föräldrar, med information om projektet samt en förfrågan om de tillät sina barn delta eller inte. Under urvalsprocessen i föreliggande studie fanns en tydlig instruktion till lärarna att de skulle informera eleverna att deltagandet i undersökningen var helt frivillig. *Konfidentialitetskravet* och *Nyttjandekravet* har beaktats på så sätt att all intervjudata har anonymiserats när det gäller person, kön och namn på skolor och klasser. Anonymiseringen har gjorts från första början, vilket innebär att den fortlöpande granskningen av studiens texter enbart innehållit anonymiserade utsagor. Alla data kommer enbart att användas i vetenskapligt syfte och nyttjandet är helt och hållet förbehållet mig och den forskargrupp som arbetat med projektet.

6. RESULTAT

Varje frågeställning har sin egen karaktär där det gäller att ha en följsamhet med och en vändning mot problemet så att fenomenen tillåts framträda och beskrivas så som de visar sig. Tranströmer uttrycker det som att ”Varje problem ropar på sitt eget språk” (1979, s.78). I föreliggande studie innebär det att försöka förstå och beskriva perspektivet i vilket sakerna – elevernas beskrivningar av programmering – visar sig. Genom en fenomenografisk analys avser jag att presentera fenomen så som de beskrivs av eleverna när jag närmat mig forskningsfrågan: *Vad riktar eleverna sin uppmärksamhet mot i sina beskrivningar av programmeringshandlingen?*

Inledningsvis presenteras några centrala delar från tidigare kapitel för att tydliggöra studiens avsikt och vad som ligger till grund för analysen. I inledningskapitlet nämnde jag att det är uppfattningar av programmeringshandlingen som är i fokus i denna studie, där intentionalitetsbegreppet används för att fånga det som är framträdande för eleverna i deras beskrivningar. Kopplingen mellan den intentionala akten och uppfattningsbegreppet utgörs av att en uppfattning alltid består av en intentional akt och att det är innehållet i beskrivningarna som är föremål i studien. För att förstå det perspektiv som eleverna beskriver har jag i analysen av data försökt att tyda de sammanflätade intentionala akterna, det vill säga hur uppmärksamheten växlar mellan flera objekt. Husserl (2002/1901) påpekar att flera akter ibland är sammanvävda, de är samtidigt givna, men att uppmärksamheten särskilt är verksam i en av dem.⁴⁰ Det är emellertid viktigt att påpeka att det inte är de enskilda tingen som är föremål för analysen. Kroksmark (1987) menar att det inte är ting som framträder i intentionaliteten, utan ett perspektiv i vilket tinget eller tingen visar sig.⁴¹ Denna skillnad mellan tinget och det perspektiv i vilket tinget visar sig beskriver Sokolowski (2000) som enkla respektive kategoriska objekt.⁴² De enkla karaktäriseras av tingen medan det kategoriska objektet är det perspektiv i vilket tinget visar sig. Sammanfattningsvis är analysen och sedermera resultatet

⁴⁰ Se kapitel 4.

⁴¹ Se kapitel 4.

⁴² Se kapitel 4.

en fråga om att det intentionala objektet framstår mot den kontextuella grund som utgörs av de samtidigt givna objekten.

KAPITLET'S DISPOSITION

Redovisningen inleds med en översiktlig beskrivning av resultatet där utfallsrummet i sin helhet presenteras på rubriknivå. Översikten följs av avsnittet *Utfallsrummet* där beskrivningskategorierna presenteras. Beskrivningskategorierna exemplifieras med hjälp av citat från intervjuerna så att innebörden i kategorierna tydliggörs. I avsnittet *Reflektioner kring utfallsrummet* förs en diskussion kring likheter och skillnader mellan beskrivningskategorierna. Kapitlet avslutas med att presentera studiens kunskapsbidrag.

Alla citat är kodade med elev och skola. Exempelvis skrivs ett citat från elev nummer två vid skola nummer tre på följande sätt: [E2S3]. De ljudenliga transkriptionerna är med försiktighet justerade språkligt för att öka läsbarheten i resultatredovisningen. Utfyllnadsljud är borttagna, och likaså dialektala ord. Tankepaus i citaten skrivs med tre punkter, och överhoppade ord eller meningar skrivs med tre punkter omgivna av hakparenteser.

RESULTATÖVERSIKT

Beskrivningskategorierna utgör studiens resultat och är uttryck för mina tolkningar av elevernas kvalitativt skilda uppfattningar. Inom ramen för studiens omfattning bör resultatet betraktas mot bakgrund av såväl intervjuens omfång som vad det innebär att tolka transkriberingar.

Analysen har resulterat i fyra beskrivningskategorier. Dessa representerar de övergripande innehållsliga aspekterna som eleverna riktar sin uppmärksamhet mot när de beskriver sina programmeringshandlingar. På nästa sida (Figur 4) visas en schematisk figur över utfallsrummets ingående kategorier och hur dessa är ordnade hierarkiskt.

- A. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot systematiskt indelning av programmet
- B. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot att komma ihåg tidigare programlösningar
- C. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot robotens rörelser
- D. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot att lösa uppgiften

Figur 4. Schematisk bild över studiens fyra beskrivningskategorier i utfallsrummet.

Avsikten med studien var att beskriva uppfattandets riktadhet, därför fanns det initialt ingen ambition att avgöra huruvida en utsaga kvalitativt innehåller en djupare förståelse av objektet programmeringshandling än en annan. Emellertid har analysen visat att utsagorna innehåller kvalitativt skilda sätt att uppfatta forskningsobjektet när det gäller förmågan att se programmeringshandlingen ur ett atomistiskt respektive holistiskt perspektiv under programmeringen. Beskrivningskategorierna är därför hierarkiskt presenterade på så sätt att de två första kategorierna, A och B, är inriktade på delar medan utsagorna under kategori C och D beskrivs med inriktning mot att försöka se helheten under programmeringen. Den innehållsliga progressionen går med andra ord från kategori A till D. En närmare beskrivning av skillnaderna och likheterna mellan beskrivningskategoriernas ges i avsnittet *Reflektioner kring utfallsrummet*.

UTFALLSRUMMET

De fyra beskrivningskategorier som visat sig i analysen av datamaterialet presenteras i detta avsnitt. Forskningsfrågans karaktär, där det intentionala objektet har en framträdande betydelse, har medfört att beskrivningskategoriernas meningsinnehåll ligger nära utsagorna. Detta kan jämföras med andra undersökningar där vida beskrivningar tillämpas och där abstraktionsnivån i kategorierna omfattar ett större innehåll (jfr. Booth, 1992; Bernmark-Ottosson, 2005). Avsikten med beskrivningskategorierna är att illustrera det perspektiv i vilket tingen visar sig och de utgör inget annat än statistiska representationer – en distansering – av dynamiska processer.

A. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT SYSTEMATISK INDELNING AV PROGRAMMET

Denna kategori kännetecknas av utsagor där det mest framträdande i beskrivningarna är programmet och hur det manipuleras med hjälp av ikonerna. Med programmet menas den programmeringsmiljö där programmeringen sker. I beskrivningarna riktar eleverna sin uppmärksamhet mot hur programmet systematiskt ställs in och en avgörande faktor för att lyckas är att dela in programmet i små sekvenser. Det som skedde under ett specifikt moment var viktigt, det som skulle komma senare var inte intressant för stunden: ”Det är väldigt svårt att tänka på det som kan hända sen efter det man gör just då. Man måste vara väldigt fokuserad på det man gör just då” (E3S1). Eleven beskriver ett arbetssätt och en strategi för att kunna lösa uppgiften på lättaste sätt och i resonemanget framträder en likhet med de problemlösningsteorier som presenterats i tidigare kapitel. Särskilt tydligt blir detta om vi jämför elevernas beskrivningar med Descartes första punkt i sin problemlösningsslagmetod:⁴³ Att dela upp ett problem i så många delar som det behövs för att lättare kunna lösa problemet (Descartes, 1983). Det vanligaste sättet att dela upp programmet/problemet var att relatera sekvenser i programmet till robotens rörelse. Om exempelvis roboten skulle åka framåt tills den stötte mot något för att sedan svänga, då beskrevs ett delmål som *tills roboten (åkt och) stöter mot något*. Därefter tog eleverna itu med den delen i programmet då roboten skulle svänga. Exemplet på nästa sida illustrerar detta – mina kommentarer inom parentes:

⁴³ Se kapitel 3.

I: Kan du beskriva hur du tänker i den här sekvensen?

E: Jag lägger först in alla de saker jag behöver ha (ikonerna placeras ut i programmet)

I: Brukar du programmera på det sättet?

E: Ja, det är lättast att lägga ut det jag behöver, innan jag binder ihop delarna. Ja, jag ställer in styrkan på motorerna också.

I: Vad är viktigt att tänka på när man programmerar?

E: Det är väldigt svårt att tänka på det som kan hända sen efter det man gör just då (pekar). Man måste vara väldigt fokuserad på det man gör just då.

I: Kan du utveckla det?

E: Man kan inte tänka att den ska svänga vänster sen, utan den måste först ta sig dit innan man kan tänka att den ska svänga vänster, för det spelar ju ingen roll nu... Den ska ju dit först. Det är viktigt att fokusera på nuet (E3S3).

Programmet/programmeringsmiljön är det centrala tinget i utsagorna som utgör denna beskrivningskategori. Det systematiska perspektiv i vilket programmet visar sig framträder dels genom den sekventiella indelningen av programmeringshandlingen som vi sett exempel på och dels genom det metodiska arbetssätt som citatet nedan exemplifierar. Min kommentar inom parentes:

I: Kan du beskriva hur du tänker i den här delen av programmet, här är du inte riktigt säker på hur du ska göra?

E: Jag visste inte riktigt hur programmet ville att jag skulle sätta ihop det här (pekar) [...] men det var ju bara och komma på att göra olika stopp.

I: Kan du beskriva det?

E: Jag tar och går från vänster till höger (pekar på ikonerna och syftar på den ordning programmet är skrivet i). Och flera gånger har jag gjort så att jag tagit en stopp i början av programmet utan att det behövs för att kolla: fungerar det här. Då kan jag göra om och om igen till det fungerar bra och sen kan jag jobba vidare med nästa del. För att veta att den kommer till rätt ställe (E1S3).

De ”stopp” som eleven beskriver är till för att göra kontroller under tiden som programmet konstrueras. Stoppen används för att isolera en del i programmet så att funktionen i en viss del kan kontrolleras. En annan elev beskrev under intervjun att det sista som tas bort innan programmet som helhet testas är stoppen som är utplacerad på olika ställen i programmet.

Gemensamt för de utsagor som ligger under denna kategori är – som tidigare nämnts – att eleverna beskriver tillvägagångssättet med att dela in programmet i mindre delar för att förenkla arbetsgången och för att inte tappa bort sig. Utsagorna innehåller ofta detaljerade beskrivningar av den funktion som ikonerna representerar och eleverna talar hellre i termer av att ställa in än att använda ordet programmera för att beskriva sin handling.

B. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT ATT KOMMA IHÅG TIDIGARE PROGRAMLÖSNINGAR

Utsagorna i denna kategori innehåller beskrivningar där uppmärksamheten är riktad mot andra programmeringslösningar som liknade den uppgift eleverna skulle utföra vid intervjutillfället. I beskrivningarna framträder att eleverna riktar sin uppmärksamhet mot att komma ihåg hur andra program varit konstruerade eller hur de själva gjort i en tidigare situation för att sedan programmera på samma sätt. Alla utsagor har gemensamt att eleverna beskriver sin egen programmeringshandling som att komma ihåg och göra på liknande sätt som de vet fungerat tidigare. Av intervjuerna att döma beskrevs återanvändningen av tidigare lösningar i de fall då eleverna stötte på något problem i programmeringen. Det var när tveksamheter uppstod som eleverna försökte komma ihåg hur de själva eller någon annan löst problemet. I det första exemplet nedan beskriver en elev hur han/hon tänkte under en sekvens i programmet när en viss tvekan uppstår. Eleven placerade en ikon på ett ställe i programmet för att sedan ta bort den och sätta den på ett annat ställe. Därefter ersätts ikonen med en annan och efter en stund tar eleven åter tillbaka den första ikonen för att slutligen använda den:

I: Här håller du på och sätter dit en bild och tar sedan bort den, och sedan sätter du dit den igen, kan du beskriva hur du tänker här?

E: Ja, jag funderade var jag skulle sätta den. Jag försökte komma ihåg var jag skulle sätta den och till slut kom jag fram till hur jag tror att det är.

I: Komma ihåg säger du, kan du utveckla det?

E: Ja, för... hur det ser ut i vilket program som helst med tid i, och var du lägger tidsbestämmelserna där.

I: Är det ett program som du själv gjort?

E: Ja, vi har gjort sådana allihop i stort sett (E1S2).

I citatet relaterar eleven till ”vilket program som helst med tid i” vilket tyder på att det inte är något specifikt program som avses utan de lösningar som eleven försöker komma ihåg är hur något program generellt såg ut som hade tid som en variabel i programmet. Detta kan jämföras med nästa citat, där ett speciellt program avses när eleven redogör för sin programmeringshandling.

Även denna utsaga illustrerar en situation där eleven stöter på ett problem och hur han eller hon kommer ihåg från ett program hur problemet kan lösas. Kommentarererna inom parentes är mina:

I: När du ser detta, hur du gjorde när du programmerade, kan du beskriva hur du tänkte när du programmerade?

E: [...] där ska den ju köra tills den stöter mot något eller i 6 sekunder och då får man ju tänka: hur var det man gör då? [...] jag kom på hur man skulle göra, jag använder en splitt eller vad man ska säga så att det fanns alternativ. Jag mindes från ett annat program att man inte fick lägga ihop de där två (pekar och syftar på att det två möjliga vägar i programmet som ikonerna för selektion innebär inte kan läggas ihop utan måste behandlas var för sig). När man gjorde antingen eller skulle den ena prioriteras före den andra (E2S1).

I jämförelse med förra citatet där eleven relaterar till ett generellt program där tidsvariabel fanns med gör problemets art att eleven i citatet ovan försöker komma ihåg ett specifikt program som innehåller samma typ av problem för att lösa uppgiften. I detta fall ska eleven lösa den del i programmet där en selektion ska ge förutsättningar för en kommandoändring hos roboten, antingen då den stöter mot något eller då motorerna gått i 6 sekunder. Det finns i datamaterialet flera exempel på att svårighetsgraden i problemet avgör om eleverna riktar uppmärksamheten mot ett specifikt program eller mot ett generellt program vilket som helst.

Det förekommer även utsagor där eleverna beskriver återanvändning av tidigare lösningar utan att de stött på några problem eller är tveksamma om hur programmeringen skulle genomföras. Dessa är inte karakteristiska för kategorin eftersom uppmärksamheten inte är riktad mot tidigare lösningar på samma sätt som de utsagor som utgör beskrivningskategorin. Det handlar snarare om att eleverna på ett erfarenhetsmässigt sätt använder sig av lösningar som de vet fungerar utan att försöka minnas hur de gjort tidigare.

C. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT ROBOTENS RÖRELSER

Kännetecknande för beskrivningar som utgör denna kategori är att eleverna riktar uppmärksamheten mot robotens rörelse och – i några fall – hur motorerna påverkar rörelsen. Rörelserna hos roboten beskrevs i de flesta fall i förhållande till uppgiftsområdet där roboten skulle utföra sina uppgifter. Det finns även utsagor där rörelserna beskrivs i förhållande till andra omgivningar än uppgiftsområdet. Det handlat om golvet eller ytan på arbetsbordet i det rum där vi befann oss under intervjun. I de utsagor som utgör denna kategori finns

RESULTAT

alltid roboten som det mest framträdande tinget under programmerings-handlingen. En elev beskriver vad som är viktigt att tänka på när man programmerar på följande sätt: ”Vad ska man säga, roboten står ju i centrum hela tiden, när du kör är det ju den som ska göra något [...]” (E2S2). Innebörden i beskrivningskategorin är emellertid ett perspektiv där inte roboten framträder som enbart en robot utan hur den rör sig och hur delarna – motorerna – påverkar rörelsen som är framträdande i utsagorna. En viktig faktor i elevernas beskrivningar är förhållandet mellan robotens rörelse och tiden. I det första exemplet uppmanades eleven att beskriva hur han eller hon tänkte under programmeringen. Beskrivningen visar att robotens (”bilen”) rörelse i förhållande till tiden är centralt:

I: Om du tänker dig in i hur du tänker när du programmerar. Hur tänkte du här när du programmerade?

E: Först måste jag veta hur jag ska köra med bilen och sen får man tänka sig in i hur lång tid det tar på ett ungefär att köra så långt. Och om den ska svänga, hur lång tid tar det att svänga, så skriver man upp det på ett papper sen att det... för att svänga 90 grader så tar det 0.6 sekunder, då har man det (E1S1).

Eleven utvecklade sitt resonemang när jag bad om en närmare beskrivning av tiderna:

E: Den kör ju en viss sträcka på en viss tid. Och då gäller det att veta att en meter tar ungefär 5 sekunder att köra, nu vet jag inte alls hur lång tid det kan ta [...] (E1S1).

Fokuseringen på robotens rörelse i förhållande till tiden var tydlig hos fler elever, men det fanns också utsagor där eleverna riktade uppmärksamheten mot robotens rörelse i generella termer så som *framåt och bakåt* och *härifrån till dit* utan att betona tidens betydelse. I exemplet nedan beskriver eleven hur han eller hon tänker sig att roboten rör sig efter programkommandon. Mina kommentarer inom parentes:

I: Kan du beskriva vad du tänkte på när du gör den här sekvensen i programmet som vi ser?

E: Ja... jag funderade på hur den skulle framåt och ifall det skulle bli ett tryck (sensor) [...] Den börjar här och åker... till där någonstans, det blir bra tänkte jag, och sedan dit [...] jag tänker mig hur roboten åker på mina kommandon (E3S3).

Under intervjun pekar eleven med fingrarna på bordet för att illustrera var roboten startar och vart den ska åka.

Det finns även utsagor där robotens motorer på får en framträdande roll. Det är främst när intervjun kommer att handla om vad eleverna uppfattar som viktigt att tänka på när de programmerar, som motorerna får en central roll för robotens rörelser:

I: Vad är viktigt att tänka på när man programmerar?

E: Det är väl att man sätter ut rätt saker så att den åker åt det hållet du vill [...]

Det är svårt att förklara hur man tänker när man själv gör det, man tänker ju så pass mycket.

I: Om du skulle försöka sätta ord på hur du tänker, vad som är viktigt?

E: Det är ju att få den kapacitet på motorerna som jag vill ha och att den åker så länge jag vill och att den svänger åt det håll jag vill (E2S3).

I exemplet ovan finns en direkt koppling mellan programmet och robotens rörelser. ”Att man sätter ut rätt saker” (Ibid.) innebär att rätt ikoner placeras ut i programmeringsmiljön, det samma gäller ”att få den kapacitet på motorerna som vill ha” (Ibid.). Emellertid är det robotens rörelse som är framträdande i beskrivningen och den påverkas av att rätt ikoner sätts ut och att motorerna får den kapacitet som krävs för rörelsen.

D. PROGRAMMERINGSHANDLINGEN BESKRIVS MED UPPMÄRKSAMHETEN RIKTAD MOT ATT LÖSA UPPGIFTEN

Utsagorna i denna kategori har gemensamt att eleverna riktar uppmärksamheten mot hur uppgiften skulle lösas med hjälp av den programmerade roboten. Till skillnad mot tidigare redovisade kategorier är ett helhetsperspektiv framträdande i utsagorna. Eleverna beskriver att de försöker skapa en helhetsbild av uppgiften genom att tänka sig inre bilder av hur roboten ska agera antingen på banan eller på golvet i rummet där intervjun äger rum. I utsagorna beskrivs programmeringshandlingen som en problemlösning där de inre bilderna hjälper eleverna att se roboten och olika hinder i förhållande till målet/syftet med uppgiften och att få en helhetsbild av den uppgift som ska utföras. Det första citatet i denna kategori illustrerar hur roboten och hindren på banan är delar i en helhet där uppmärksamheten är riktad mot hur uppgiften ska lösas:

RESULTAT

I: Kan du beskriva vad det är du tänker på när du programmerar?

E: Jag tänker... först tänker jag på vad är det jag ska lösa. I detta fall skulle jag köra fram roboten tills den stöter emot. Då får jag se vad som finns [...] och använder jag för instrument för att få den att stanna [...].

I: Du beskrev tidigare att du tänker på vad är det jag ska lösa, hur tänker du på det?

E: Nu är det ju inga hinder som man vet om eller något i vägen, utan den ska köra framåt för att sedan åka bakåt och svänga. Då får man ju liksom bygga upp en bild i huvudet, vad är det som ska hända och hur.

I: Hur ser den bilden ut?

E: I detta fall, så kan man tänka sig ett kalt... platt golv med en vägg i vägen som den ska köra rakt in i, det är ett exempel [...] (E2S1).

I nästa exempel beskriver eleven att det är uppdraget som framträder under programmeringshandlingen och vad han eller hon ”vill göra med bilen” (E2S2). Utsagan vittnar om att eleven känner kontroll i situationen eftersom det är han eller hon som avgör vad bilen ska göra för att uppdraget ska bli utfört:

I: Hur tänker du när du programmerar under denna sekvens?

E: Det är väl på mitt uppdrag eller uppgift hela tiden, jag tänker på vad jag vill göra med bilen, och nu får jag ju här att den ska köra framåt i visst antal sekunder och sedan backar, och då tänker jag mig det. Man får ju tänka sig på ett ungefär som man kör mellan stolsbenen, hur stor den är och hur man har drevat och hur lång tid det på ett ungefär kommer att ta (E2S2).

Som framgår av citatet beskriver eleven att han/hon tänker sig att roboten kör mellan stolsben. Eleven återkommer till detta resonemang under en senare del av intervjun och tydliggör sin beskrivning av den inre bilden genom att tala om, att roboten kör på golvet och trycksensorn aktiveras då roboten kör in i ett stolsben.

Beskrivningskategorin bärs upp av två betydande aspekter. För det första förmågan att skapa den mentala bild som eleverna beskriver och för det andra att det är uppgiften som framträder i den mentala bilden. Några elever menar att den mentala bilden är något som man måste träna sig till, att man måste över en viss gräns för att kunna tänka på detta sätt, och att de som har förmågan att tänka på detta sätt tänker rätt. I citatet nedan framgår dessa tankegångar. Eleven beskriver vad man bör tänka på när man lär sig programmering:

E: Vad du bör tänka på... det är två sätt i programmering, ett sätt är att först kunna språket, kommer du över den gränsen och har orken då gäller det att kunna det andra med bilden, du ska kunna skapa dig en bild och se lösningar på problemen [...] vill lära dig att programmera gäller det att tänka rätt, kan du tänka rätt tror jag man har förutsättningar för det.

RESULTAT

I: Hur tänker man rätt då?

E: Det tror jag är en liten tränings sak, som det jag var inne på med bilden, att du ska få dig en bild och att du ska plocka in faktorerna och så där. Man får testa så får man mer kunskap[...] (E3S3).

Förmågan att skapa mentala bilder är enligt eleven en tränings sak där det handlar om att först lära sig programmet. Detta tyder på att kunskap om programmet är viktigt för att komma vidare och att kunna se uppgiften genom mentala bilder är en skicklighetsnivå över programkunskaper. Det senare är påtagligt i och med att eleven beskriver detta som ett av två sätt att programmera. Antingen stannar man vid att kunna enbart programmet eller så har man ”orken” (Ibid.) att gå vidare och finna lösningar på problemen genom att skapa inre bilder.

REFLEKTIONER KRING UTFALLSRUMMET

I detta avsnitt diskuteras några utmärkande aspekter i utfallsrummet. Den inledande diskussionen tar sin utgångspunkt i beskrivningskategorierna där en gruppering görs för att beskriva den hierarkiska ordningen.

- A. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot systematiskt indelning av programmet
- B. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot att komma ihåg tidigare programlösningar
- C. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot robotens rörelser
- D. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot att lösa uppgiften

De fyra beskrivningskategorierna som utgör resultatets utfallsrum kan grovt delas in i två grupper. Den första gruppen innehåller kategori A och B medan den andra gruppen utgörs av kategori C och D. Den första gruppen utgörs av elever som riktar sin uppmärksamhet mot *delar* i problemlösningssprocessen när de beskriver sin programmeringshandling. Vid en jämförelse mellan kategori A och B finner vi likheter i beskrivningarna när det gäller att lyfta fram delar framför helheten i programmeringsmomentet. I flera utsagor beskriver eleverna att de delar in problemet i delmål, som löses var för sig, där var och ett ska klaras av innan nästa moment i programmeringen tar vid. Detta gäller såväl

RESULTAT

det systematiska tillvägagångssättet (A) som att komma ihåg tidigare programlösningar (B). Med andra ord är detaljrikedomen utbredd i utsagorna och på den punkten liknar kategorierna varandra. De utsagor som utgör kategori A och B, och som i denna diskussion bildar den första gruppen, kan beskrivas som att eleverna har ett atomistiskt förhållningssätt till sin programmeringshandling. Med atomistisk menar jag att elevernas beskrivningar ofta handlar om enskilda objekt där programmeringshandlingen innebär att problemet delas upp och delarna löses var för sig för att sedan sättas ihop till en hel lösning bestående av summan av delarna.

Utsagorna i den andra gruppen, som utgörs av kategorierna C och D, är i jämförelse med den första gruppen mer inriktade mot ett helhetsperspektiv i beskrivningarna. Eleverna försöker se hela problemlösningsprocessen när de beskriver programmeringshandlingen och utsagorna ansluter till ett mer holistiskt perspektiv än i den första gruppen. Med utgångspunkt i denna gruppering, atomistiska respektive holistiska beskrivningar av programmeringshandlingen, är de utsagor som finns under andra gruppen kvalitativt mer utvecklade i jämförelse med den första gruppen när det gäller förmågan att förstå programmering och att lösa uppgiften. Det finns alltså en hierarki mellan grupperna och följaktligen också mellan beskrivningskategorierna. Den hierarkiska ordningen går från kategori A till D där utsagorna i kategori D är kvalitativt mer utvecklade än de övriga och innehåller C, B och A. C innehåller B och A och så vidare.

Det är emellertid viktigt att påpeka att det från intervjumaterialet är svårt att avgöra huruvida utsagorna i kategori B – [...] *att komma ihåg tidigare programlösningar* – avser att komma ihåg lösningar som enskilda objekt för att använda dessa eller om eleverna försöker komma ihåg hur de tänkte då de gjorde en lösning. Handlar det om det senare kan det knappast ordnas under ett atomistiskt förhållningssätt utan det blir då ett sätt att använda tidigare erfarenheter. När det gäller kategori A – [...] *systematisk indelning av programmet* – kan indelningen av programmet, till mindre isolerade delar som ska lösas, vara en produkt av att eleverna i de flesta fall arbetar i grupper. Ett vanligt arbetsätt i grupperna är att dela upp programmeringen mellan eleverna i mindre delar som sedan sätts ihop till ett program. Detta kan vara en orsak till att några elever ser detta som ett naturligt sätt att arbeta. Likväl ser jag inte att detta kan påverka studiens resultat eftersom uppgiften som eleverna fick att lösa inte var omfattande. Den kan mycket väl liknas med ett mindre moment i ett större

program och eleverna riktade uppmärksamheten mot att systematiskt dela upp programmet trots att uppgiften inte ingick i ett större program.

Utfallsrummets struktur påminner om Booths (1992) resultat på så sätt att hon fann fyra kvalitativt skilda kategorier av hur studenter uppfattar att de lär sig programmera ett rekursionsproblem⁴⁴ samt att dessa kan delas in i två grupper. De två första kategorierna i Booths resultat visar att studenterna är fokuserade på programmeringsspråket eller själva syntaxen. Kategori tre och fyra pekar på något bortom själva instruktionerna till datorn, dessa har fokus på hur problemet skulle lösas med hjälp av programmeringen.

FÖRMÅGAN ATT SKAPA INRE BILDER

De elever, som i sina utsagor riktar uppmärksamheten mot att lösa uppgiften, är kvalitativt mest utvecklade när det gäller förmågan att förstå och beskriva programmeringshandlingen. Gemensamt för dessa utsagor är att eleverna beskriver hur de skapar mentala bilder för att se hur roboten ska röra sig så att uppgiften blir utförd. Under intervjuerna har det också framkommit att de inre bilderna är något som man måste träna sig till, att man måste över en viss gräns för att kunna tänka på detta sätt (se citat sidan 84). Eleverna beskriver med andra ord detta som en förmåga och uppnås detta hjälper de mentala bilderna eleverna att lägga nya erfarenheter till tidigare genom att ”bilderna uppdateras”(E2S1). I exemplet nedan beskriver en elev vad som menas med en uppdatering. Eleven beskriver hur han/hon på förhand kan se att roboten kommer att spinna på grund av att underlaget är löst:

E: [...] då kan roboten spinna till exempel. Och då vet jag det sen tidigare kunskaper ... som en bild ... som jag använder i grund och botten för den här världen. Sen kan de ju vara så att hindren inte är där som jag tänker, då märker jag antagligen det när jag provkör roboten. Och då får man göra en annan sorts bild, då får jag uppdatera min bild.

I: Hur kan en sådan här uppdatering gå till då?

E: Vad ska man säga...om du tänker att du går ut i naturen på ett berg ... och sen lyfter jag upp en sten som jag tror väger jättemycket, men sen var den visst lätt och då, jaha då var den där lätt då vet jag det. Då ser jag på den hur den ser ut och då försöker jag komma ihåg det, sådana som ser ut så, sådana är lätta stenar (E2S1).

Eleven hänvisar till tidigare kunskaper när bilden beskrivs och hur den uppdateras. Detta är något som återkommer i fler utsagor i den sista kategorin –

⁴⁴ Se kapitel 2.

RESULTAT

[...] *uppmärksambeten riktad mot att lösa uppgiften* – och framträdande är att den mentala bilden är beroende av tidigare erfarenheter. I citatet ovan är elevens redogörelse för hur en uppdatering går till slående och exemplet med att gå ut i naturen för att lära sig hur en stens vikt förhåller sig till storleken, tyder på en förmåga att kunna relatera ett problem i ett sammanhang till ett annat sammanhang med ett vidgat perspektiv. En annan intressant del i utsagan, som bör uppmärksammas, är att i citatet heter det: ”för den här världen”(E2S1), vilket vittnar om att den mentala bild som eleven beskriver ses som en aspekt i livsvärlden, det vill säga i ett sammanhang där arbetet med programmering och problemlösning äger rum.

OBSERVATIONER UNDER INTERVJUERNA

En viktig del inom livsvärldsfenomenologisk forskning är att delta på plats där studien äger rum. I föreliggande studie har min medverkan inneburit att jag lättare kunnat förstå vad det är eleverna uttrycker och vad som är viktiga moment i arbetet med programmeringen. Under studien har jag besökt klasser och elever som dels deltagit som informanter och dels arbetat med materialet i undervisningen utan att ha någon koppling till undersökningen. Besöken har möjliggjort att jag kunnat vara närvarande i undervisningen och observerat hur eleverna arbetat med materialet och vad som är centralt för eleverna under deras aktiviteter. Bland annat har den aktiva närvaron hjälpt mig att förstå det sammanhang som jag och eleverna befinner oss i, och gemensamt bidrar till, och vilka förutsättningar som påverkat programmeringsaktiviteten för eleverna. Två aspekter har visat sig när det gäller saker som påverkat elevernas sätt att programmera och arbeta med materialet. Den första är tillgången till batterier och den andra aspekten är att robotens konstruktion har en avgörande betydelse.

Tillgången till batterier var begränsad i klasserna på grund av kostnaden. Detta medförde att eleverna hittade strategier för att programmera och testa robotarna på ett sätt som sparade på batterikapaciteten. Ett exempel på detta var då motorerna gick åt fel håll så korrigerades detta med att eleverna vände på sladdarna som försåg motorerna med ström istället för att göra ett nytt program. I och med att sladdarna vändes skiftades polerna och motorerna gick åt andra hållet. Enligt eleverna är detta både ett snabbt och energisnålt sätt att rätta till riktningen på motorerna. På samma tema har det förekommit diskussioner i grupperna under arbetets gång om metoden trial-and-error ska användas eller inte eftersom varje testning kräver batteriresurser. Vid ett tillfälle

RESULTAT

uppmnade en elev de andra i gruppen att ”var och en får göra tre tester” (Observation, september 2005) annars menade eleven att det slösades på batterier. Till saken hör att det i några klasser var överdrivet fokus på batterierna och hur kapaciteten rättvist skulle fördelas inom gruppen. Vid ett samtal med lärare i dessa klasser visade det sig att tillgången på batterier inte var dålig, men att de ville lära eleverna att hushålla för att de inte skulle byta batterier ”så fort som roboten inte går dit de vill” (Ibid). I de klasser som jag besökt verkade likväl batteritillgångens ha en stor betydelse för eleverna som påverkande aspekt då robotarna programmerades.

Den andra aspekten är att robotens konstruktion har haft en avgörande betydelse för programmeringsmomentet. Bland annat måste den hålla för den eller de uppgifter som ska utföras, och programmeraren måste veta vilka sensorer etc. som finns på roboten för att avgöra vilka funktioner som kan användas vid programmeringen. En annan aspekt som visat sig under programmeringsmomentet, och som påverkade elevernas resultat, var att de elever som själva konstruerade den robot som de sedan programmerade hade enklare att rätta till de fel som uppstod vid programmeringen. Eftersom eleverna hade konstruerat roboten själva kände de till alla delar och flera fall visade på att konstruktionen ofta gjordes med både uppgiften och det kommande programmet i åtanke. Exempelvis ersattes ljussensorn med några extra trycksensorer på roboten så att den kunde orientera sig och utföra uppdraget. Motivet var att trycksensorn är lättare att programmera än ljussensorn. Det enskilda arbetet, där samma elev gjorde såväl robotbygget som programmeringen, visade på ett bättre resultat och i flera fall en djupare förståelse för programmeringsmomentet jämfört med de elever som ingick i grupper där de delade upp arbetet mellan sig. Uppdelningen av arbetet i grupperna innebar ofta att någon byggde chassi, några andra elever byggde de verktyg och armar som roboten behövde för att utföra uppgiften och ytterligare en annan programmerade konstruktionen. Enligt flera lärare fanns det i det här fallet ingen pedagogisk tanke med att låta eleverna arbeta i grupper utan det var på grund av bristande tillgång på byggsatser (Lärare skola 1, Lärare skola 3). De elever som haft möjligheten att arbeta självständigt har gjort det i de fall då lektionerna varit i halvklass och tillgången på byggsatser varit god.

KUNSKAPSBIDRAG

Resultatet i denna studie utgörs av fyra olika sätt att rikta uppmärksamheten när programmeringshandlingen beskrivs, vilka tillsammans utgör ett utfallsrum av möjliga intentionala objekt. Strukturen i detta utfallsrum har bedömts bestå av två inriktningar, nämligen atomistiskt respektive holistiskt perspektiv. I kapitlets inledning nämnde jag att det initialt inte fanns någon ambition att avgöra huruvida en utsaga kvalitativt innehåller en djupare förståelse av objektet programmeringshandling än en annan. Emellertid visade analysen att de fyra kategorierna kunde delas in i två inriktningar. Jag vidhåller trots det att studier av detta slag, där uppmärksamhetens riktadhet studeras, bör hantera ett hierarkiskt mönster med försiktighet. Anledningen är att det knappast går att göra anspråk på att avgöra huruvida ett intentionalt objekt är kvalitativt mer utvecklat eller innehåller en djupare förståelse för programmeringshandlingen än ett annat. Likväl visar resultatet på ett möjligt sätt att hierarkiskt dela in forskningsobjektet, vilket medför att denna diskussion om studiens kunskapsbidrag tar sin utgångspunkt i det som benämns ett holistiskt lärande (Marton & Booth, 2000).

Inom fenomenografin ser man en del lärande som effektivare än andra och man tar ställning för en holistisk inriktning i undervisningen (Marton & Booth, 2000). Genom att lärare har en inblick i och förståelse för elevers olika sätt att uppfatta lärandeobjekten har de möjlighet att skapa förutsättningar för lärandet. Genom resultatet i föreliggande studie är avsikten att bidra till att lärare lyckas skapa en dynamisk undervisning där det ges utrymme för de kvalitativt skilda sätt att erfara centrala aspekter under programmeringshandlingen som framgår av utfallsrummet. Elevernas sätt att tänka under programmeringen utgör ramen för den arbetsmetod som eleverna själva ser som naturlig och mest effektiv för att lyckas med uppgiften. Om läraren har förståelse för elevers olika sätt att rikta uppmärksamheten i samband med programmeringen och tar detta som didaktisk utgångspunkt underlättas ett holistiskt lärande. Poängen med detta är inte att förespråka en metod eller ett utfallsrum som ska tas för sant i positivistisk mening, utan ett sätt att fånga så många elever som möjligt med utgångspunkt i troliga sätt att tänka under programmeringsmomentet.

RESULTAT

I utfallsrummet finns det två kategorier som ordnats under holistiskt perspektiv: *C. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot robotens rörelser* och *D. Programmeringshandlingen beskrivs med uppmärksamheten riktad mot att lösa uppgiften*. Gemensamt för dessa är att eleverna beskriver att de skapar mentala bilder där de försöker se antingen robotens rörelse eller hur uppgiften ska lösas. Utsagorna i dessa kategorier visar på att det är viktigt för eleverna att ha en miljö att relatera till där uppdraget ska utföras för att kunna se helheten. Eleverna refererar till hinder och andra fasta föremål på uppgiftsområdet eller på golvet i rummet där de programmerade när de beskriver hur de mentala bilderna hjälper dem med robotens position i förhållande till programmeringsmomentet. Med kategori C och D som utgångspunkt är det viktigt att försöka möta eleverna i deras mentala bilder och ha förståelse för att dessa kan gälla såväl närmiljön – uppgiftsområdet eller någon annan bana där uppgifterna ska utföras – som någon miljö utanför klassrummet. Utsagorna tyder också på att det är viktigt att det finns ett fysiskt uppgiftsområde med olika fasta hinder eller uppdrag som kan fungera dels, som en gemensam referenspunkt i elevernas diskussion och dels, som hjälp för de elever som skapar mentala bilder för att kunna se hela uppdraget framför sig under programmeringsmomentet.

7. DISKUSSION

Bakgrunden till föreliggande studie är en undran över hur elever lär sig programmera. Genom projektet *Lärande och programmerbart konstruktionsmaterial* gjordes en ansats till att studera aspekter av lärande och programmering, vars resultat presenterats i denna uppsats. I projektet avgränsades mina generella funderingar till att studera hur och mot vad en grupp elever med ett specifikt programmeringsverktyg riktade sin uppmärksamhet under programmeringen. Målet med avgränsningen var att hitta aspekter av programmeringsområdet som inte tidigare studerats och på så sätt bidra till kartläggningen inom området. Programmering är ett undervisningsmoment som av många – både lärare och elever – upplevs som komplicerat. Min egen erfarenhet, som lärare på universitetsnivå och genom föreliggande studie, säger mig att lärande av programmering i stor utsträckning handlar om trial-and-error metodik både från lärarnas och från elevernas sida. Lärarna vill att eleverna ska våga prova sig fram och då ligger tillvägagångssättet nära. Från eleverna perspektiv är trial-and-error en effektiv metod att komma framåt i de uppgifter som de har för handen eftersom de flesta programmeringsmiljöer erbjuder en tydlig guide som visar vad som är fel i programmet. Eleverna korrigerar de delar som felmeddelandet visar och försöker igen. I de skolor som deltagit i studien har eleverna i stor utsträckning fått lära sig programmeringsmomentet själva. Detta lärande har skett antingen genom att prova sig fram eller genom att eleverna lär varandra. En elev uttrycker det så här:

I: Har ni pratat med varandra i klassen om hur man kan tänka när man programmerar?

Elev: Nej, inte så mycket... men alltså om man behöver hjälp med något som man inte förstår hur man ska göra, måste man ju förklara så att den andre förstår vad det är man gjort, inte bara visa att du ska ställa upp den så här [...] även om det inte är så lätt att förklara.

I: Kan du se på den andre att den förstår?

Elev: Ja, det syns på ansiktsuttrycket (E3S3).

Det finns många positiva aspekter med att eleverna själva får söka efter lösningar som ger dem erfarenheter och som utvecklar deras kunskaper inom området. Likväl påtalade flera elever att de saknade hjälp från lärarna för att komma vidare i sina programmeringskunskaper och flertalet lärare ansåg att de

hade för lite kunskap om hur programmeringsundervisning går till för att känna sig trygga i undervisningssituationen. Att eleverna har svårt att dra paralleller till andra områden än det som är i den omedelbara närheten utan hjälp från läraren, har också visats i andra studier (Ivarsson, 2002). Förhoppningsvis bidrar föreliggande studie till att vidga lärarnas förståelse för hur programmering kan uppfattas av eleverna, vilket sedan även bör höja kvaliteten i programmeringsundervisningen.

REFLEKTIONER KRING SYFTE OCH METOD

Svårigheten med att studera hur någon annan tänker i en viss situation, har gjort sig påmind vid flera tillfällen under studien. Under intervjuerna har det varit påtagligt att eleverna inte är vana vid den typen av frågor som ställdes. Att konfronteras med frågeställningar som: Hur tänker du under detta moment? har i flera fall varit obekvämt för eleverna. I några fall krävdes det att frågan eller frågorna återkom vid flera tillfällen under intervjun för att eleverna skulle reflektera över och beskriva hur de tänkte under en situation som de vanligtvis inte behöver verbalisera.

En återkommande tanke under studien har varit huruvida elevernas beskrivningar refererar till samma objekt, vilket är en förutsättning för att vi ska kunna tala i termer av kvalitativt skilda sätt att erfara ett objekt. Vid en jämförelse med andra fenomenografiska studier så finns det i de flesta fall ett tydligt objekt – som kan betraktas som konstant – vilket informanterna beskriver sin uppfattning av. I denna studie kan å ena sidan varje elevs programmeringshandling betraktas som ett objekt, vilket innebär att det funnits nio olika objekt. Å andra sidan skulle alternativet vara att jag eller någon annan hade skapat ett program som eleverna skulle beskriva. Det uppenbara blir i det sistnämnda att forskningsfrågan förflyttas till vad eleverna riktar sin uppmärksamhet mot när de beskriver ett program, istället för vad de riktar sin uppmärksamhet när de beskriver sin egen programmeringshandling. Frågan om huruvida eleverna beskriver samma objekt eller inte är viktig, inte minst ur ett forskningsmetodiskt perspektiv. Den fenomenografiska ansatsen har, trots en urskiljbar objektsvariation, varit lämpligt analysverktyg för studiens forskningsfråga. Ambitionen var att beskriva hur elever uppfattar ett specifikt kunskapsinnehåll, i utbildningssammanhang benämns detta ofta som hur elever tänker, och resultatet har visat på fyra kvalitativt skilda sätt att uppfatta programmeringshandlingen.

I uppsatsens syfteformulering deklarerades ambitionen med att bidra till utvecklingen av ett erfarenhetsfilosofiskt perspektiv inom den teknologiska disciplinen. Med utgångspunkt i livsvärldsfenomenologin har bland annat betydelsen av människans erfarenheter påvisats när det gäller hur vi förstår teknik. Det gäller inte minst programmering. I utsagorna kan vi se exempel på hur eleverna upplever programmeringsmomentet som en del i världen, en aspekt av det gemensamt upplevda, och hur erfarenheter i en situation kan relateras till andra situationer i vardagen.⁴⁵ Utgångspunkten i studien har varit elevernas erfarenhet av företeelsen att programmera i det gemensamt upplevda sammanhanget. Eleverna har beskrivit hur de uppfattar programmeringsmomentet, men samtidigt har jag upplevt sammanhanget med mina erfarenheter vilket bidragit till min förståelse för programmeringens intentionala objekt hos nio elever i skolår 8 och 9.

Kunskapsbidraget ska förhoppningsvis inte enbart komma till användning i de skolor som arbetar med programmerbart konstruktionsmaterial, utan även i programmeringsundervisning inom såväl grundskolan som gymnasium och universitet. Ambitionen är att resultatet dels kan användas i undervisningen som en teoretisk utgångspunkt för vidare diskussioner om lärande av programmering och dels som ett stöd för de lärare som praktiskt undervisar inom området.

⁴⁵ Se citaten på sidorna 84 och 87.

REFERENSER

- Abelli, B. (2004). *Programmeringens grunder - med exemplet C#*. Lund: Studentlitteratur.
- Alerby, E. (2000). Lärande - några betraktelser från olika horisonter. I E. Alerby, P. Kansanen & T. Kroksmark (Red.), *Lära om lärande* (s. 17-28). Lund: Studentlitteratur.
- Alexandersson, M. (1994). *Metod och medvetande*. Göteborg: Göteborgs universitet, ACTA.
- Alexandersson, M. (1996). *Den konstnärliga blicken - en studie om konstnärers reflektion*. Göteborgs universitet: Institutionen för metodik Nr. 10.
- Backhouse, R. (2003). *Program Construction - Calculating Implementations from Specifications*. The University of Nottingham, UK: Wiley.
- Becker, P. (2000). Hur fungerar det här med IT egentligen i klassrummet, när fungerar det - och när fungerar det inte? *Datorn i utbildningen*, Nr. 7.
- Bengtsson, J. (1987a). *Edmund Husserls Filosofi - En introducerande översikt över hans fenomenologi och dess inflytande*. Göteborgs universitet: Sociologiska institutionen Nr. 90.
- Bengtsson, J. (1987b). *Maurice Merleau-Pontys Filosofi - Några grunddrag*. Göteborgs universitet: Sociologiska institutionen Nr. 91.
- Bengtsson, J. (1993). *Sammanflätningar - Husserls och Merleau-Pontys fenomenologi*. Göteborg: Daidalos.
- Bengtsson, J. (1998). *Fenomenologiska utflykter*. Göteborg: Daidalos.
- Bengtsson, J. (1999). En livsvärldsansats för pedagogisk forskning. I J. Bengtsson (Red.), *Med livsvärlden som grund* (s. 9-49). Lund: Studentlitteratur.
- Berger, P., & Luckmann, T. (1967). *The Social Construction of Reality*. London: Penguin Books.
- Berglund, A. (2005). *Learning computer systems in a distributed project course - The what, why, how and where*. Uppsala: Uppsala universitet Nr. 62.
- Bernmark-Ottosson, A (2005) *Demokratins stöttepelare - En studie av lärarstuderandes demokratiuppfattningar*. Karlstad University Studies 2005:7.
- Bjurwill, C. (1995). *Fenomenologi*. Lund: Studentlitteratur.
- Booth, S. (1992). *Learning to program - A phenomenographic perspective*. Göteborgs universitet, ACTA, Göteborg.

REFERENSER

- Calderhead, J. (1981). Stimulated Recall: A Method for Research on Teaching. *British Journal of Educational Psychology*, Nr 51, (s.211-217).
- Carlsson, S. (2000). *Lärande systemutveckling och samarbetsformer - Från kommunikation till samförstånd genom lärande och undervisning*. Karlstad: Karlstads universitet Nr 2000:01.
- Dahlin, B. (1989). *Religionen, själen och livets mening - En fenomenografisk och existensfilosofisk studie av religionsundervisningens villkor*. Göteborgs universitet, ACTA, Göteborg.
- Dahlin, B. (2002). *Den tunga vetenskapen - Lärarstuderandes uppfattningar av naturvetenskap med kontroversen mellan Goethes och Newtons optik som utgångspunkt*. Karlstad: Karlstads universitet Nr. 2002:7
- Descartes, R. (1983). Ur avhandling om metoden - Andra delen (K. Marc-Wogau, Övers.). I *Filosofin genom tiderna, 1600-talet och 1700-talet - Texter i urval* (s. 46-51). Stockholm: Bonniers.
- Dewey, J. (1998/1910). *How We Think - A Restatement of the Relation of Reflective thinking to the Educative Process*. Boston, USA: Houghton Mifflin Company.
- diSessa, A. A. (2000). *Changing Minds - Computers, Learning, and Literacy*. Massachusetts, USA: MIT Press.
- Dix, A. J., Finlay, J. E., Abowd, G. D., & Beale, R. (1998). *Human-Computer Interaction*. London: Prentice Hall Europe.
- Dreyfus, H. L., & Dreyfus, S. E. (1986). *Mind over Machine. The Power of Human Intuition and Expertise in the Era of the Computer*. New York: The Free Press.
- Eklund, S., & Fernlund, H. (1998). *Programkonstruktion med kvalitet – projekthantering och ISO 9000*. Lund: Studentlitteratur.
- Englund, T., & Svingby, G. (1986). Didaktik och läroplansteori. I F. Marton (Red.), *Fackdidaktik Volym 1 - Principiella överväganden, Yrkesförberedande ämnen* (s. 97-152). Lund: Studentlitteratur.
- Filosoflexikonet*. (1997). Stockholm: FORUM.
- Goldkuhl, G., & Röstlinger, A. (1988). *Förändringsanalys - Arbetsmetodik och förhållningsätt för goda förändringsbeslut*. Lund: Studentlitteratur.
- Griffin, R. E., & Gibbs, W. J. (1993). *International Icon Symbols: How Well Are These Symbols Understood?* Art, science and visual literacy; selected readings from the 24th Annual Conference of the International Visual Literacy Conference, (s.132-139).
- Göranzon, B., Gullers, P., Mäkilä, K., Svensson, P., & Thollander, L. (1983). *Datorn som verktyg - Krav och ansvar vid systemutveckling*. Lund: Studentlitteratur.
- Haglund, B. (2003). Stimulated Recall - Några anteckningar om en metod att generera data. *Pedagogisk Forskning i Sverige*, Årg 8, Nr. 3, (s.145-157).

- Haglund, D. (1989). Husserl och den fenomenologiska filosofin (J. Bengtsson, Övers.). I *Fenomenologins idé* (s. 9-32). Göteborg: Daidalos.
- Hanly, J., & Koffman, E. (1999). *Problem Solving & Program Design in C*. Massachusetts, USA: Addison-Wesley.
- Heidegger, M. (1974/1962). *Teknikens väsen - och andra uppsatser* (R. Matz, Övers.). Udevalla.: Rabén&Sjögren.
- Heidegger, M. (1992/1927). *Varat och tiden*. Band 1 och 2 (R. Matz, Övers.). Göteborg: Daidalos.
- Husserl, E. (1989). *Fenomenologins idé* (J. Bengtsson, Övers.). Göteborg: Daidalos.
- Husserl, E. (2000/1900). *Logiska undersökningar*. Andra bandet (K. Weigel, Övers.). Stockholm: Thales.
- Husserl, E. (2002/1901). *Logiska undersökningar*. Tredje bandet (J. Jakobsson, Övers.). Stockholm: Thales.
- Hägerfors, A. (1995). *Att samlära i systemdesign*. Lund: Studentlitteratur.
- Høyer, R. (1979). *Systemutveckling som en lärandeprocess inom företaget: Några pedagogiska krav på systemutvecklingsmodeller*. Göteborgs universitet och Chalmers tekniska högskola: Inst. informationsbehandling-ADB.
- Ivarsson, J. (2002). Tala, peka och lära matematik i datorbaserade miljöer: En kritisk analys. I R. Säljö & J. Linderöth (Red.), *Utm@ningar och e-frestelser - it och skolans lärkultur* (s. 59-76). Stockholm: Prisma.
- Jacquette, D. (2004). Brentano's concept of intentionality. I D. Jacquette (Red.), *The Cambridge Companion to Brentano* (s. 98-130). Cambridge: Cambridge University Press.
- Kant, I. (1952/1781). *The Critique of Pure Reason* (J. M. D. Meiklejohn, Övers.). London: William Benton Publicer: Encyclopedia Britannica.
- Kemp, P. (2005). *Världsmiddagaren - Politisk och pedagogisk filosofi för det 21 århundradet*. Göteborg: Daidalos.
- Kindborg, M. (2003). *Concurrent Comics - Programming of social agents by children*. Lindköpings universitet, Avd. Computer and Information Science.
- Klassner, F., & Anderson, S. D. (2003, june 2003). LEGO Mindstorms: Not just for K-12 Anymore. *IEEE Robotics & Automation Magazine*, June 2003.
- Kroksmark, T. (1987). *Fenomenografisk didaktik*. Göteborgs universitet, ACTA, Göteborg.
- Kroksmark, T. (2002) Pedagogiskt arbete. Avläst 2005-11-04, från:
<http://www.hlk.hj.se/doc/2433>
- Kroksmark, T. (2003). *Den tidlösa pedagogiken*. Lund: Studentlitteratur.
- Kroksmark, T. (2006) Innovativt lärande. *Didaktisk Tidskrift*, Vol 16, Nr 2-3. Jönköping University Press

REFERENSER

- Kurtland, D. M., Clement, C. A., Mawby, R., & Pea, R. D. (1987). Mapping the cognitive demands of learning to program. I K. Sheingold & R. D. Pea (Red.), *Mirrors of minds. Patterns of experience in educational computing* (s. 103-127). Norwood: Ablex.
- Kvale, S. (1997). *Den kvalitativa forskningsintervjun*. Lund: Studentlitteratur.
- Langefors, B. (1995). *Essays on Infology*. Lund: Studentlitteratur.
- Liedman, S.-E. (2002). *Ett oändligt äventyr - Om människans kunskaper*. Viborg, Danmark: Bonnier.
- Lilja, P., & Lindström, B. (2002). Vad ska man ha den till då? Om konstruktionistisk teknologi och lärande i skolans värld. I R. Säljö & J. Linderöth (Red.), *Utm@ningar och e-frestelser - it och skolans lärkultur* (s. 33-58). Stockholm: Prisma.
- Lindblad-Gidlund, K. (2005). *Techno Therapy - a relation with technology*. Umeå: Umeå universitet 2005:2000898.
- Lindh, J., Carlsson, S., Kilbrink, N., & Segolsson, M. (2003). *Programmerbart konstruktionsmaterial i undervisningen* (Slutrapport). Jönköping: Högskolan för lärande och kommunikation.
- Marton, F., Dahlgren, L. O., Svensson, L., & Säljö, R. (1977). *Inläring och omvärldsuppfattning*. Stockholm: Almqvist & Wiksell Förlag AB.
- Marton, F. (1981). Phenomenography - Describing conceptions of the world around us. *Instructional Science*, 10, (s.177-200).
- Marton, F. (1986). Vad är fackdidaktik? I F. Marton (Red.), *Fackdidaktik: Volym 1 - Principiella överväganden, Yrkesförberedande ämnen* (s. 15-78). Lund: Studentlitteratur.
- Marton, F., & Booth, S. (2000). *Om lärande*. Lund: Studentlitteratur.
- Merleau-Ponty, M. (2002/1945). *Phenomenology of Perception* (C. Smith, Övers.). London: Routledge.
- Moran, D. (2000). *Introduction to Fenomenology*. London: Routledge.
- Nordin, P., & Wilde, J. (2003). *Humanoider – självlärande robotar och artificiell intelligens*. Stockholm: Liber.
- Papert, S. (1980). *Mindstorms - Children, Computers, and Powerful Ideas*. Brighton: Harvester Press.
- Papert, S. (1993). *The Children's Machine - Rethinking School in the age of the Computer*. New York: BasicBooks/HarperCollins Publishers.
- Papert, S. (1994). *Hur gör giraffen när den sover? - Skolan, datorn och kunskapsprocessen*. Göteborg: Diados.
- Papert, S., Iturizaga, I. M., & Aaron, F. (1999). *Educational Impact of LEGO Dacta Materials: INFOESCUELA - MED*. Peru.

- Papert, S. (2003). *Ghost In The Machine*. Avläst 2003-09-18, från:
<http://papert.org>
- Parent, M., & Laurgeau, C. (1984). *Robot Technology - Logic and Programming*.
 Lund: Studentlitteratur.
- Pettersson, R. (2003). *Ord, bild & form - termer och begrepp inom informationsdesign*.
 Lund: Studentlitteratur.
- Pólya, G. (1990/1945). *How to Solve It - A New Aspect of Mathematical Method*.
 England: Penguin book.
- Preece, J. (1994). *Human-Computer- interaction* (First ed.). Wokingham, England:
 Addison-Wesley.
- Schütz, A. (1999). *Den sociala världens fenomenologi* (S. Andersson & J. Retzlaff,
 Övers.). Göteborg: Daidalos.
- Schön, D. (2003). *The Reflective Practitioner - How professionals think in action*.
 Aldershot : Ashgate.
- Segolsson, M. (2005). *Pupils Experience of Programming Computers - An Example of
 Learning with Computers from Swedish Compulsory School*. Paper presented at the
 ECER Dublin Congress 2005-09-08, Dublin.
- Shneiderman, B. (1986). Empirical Studies of Programmers: The Territory,
 Paths, and Destinations. I E. Soloway & R. Iyengar (Red.), *Empirical Studies
 of Programmers*. Norwood: Ablex Publishers.
- Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface* (4th ed.).
 Boston, USA: Addison-Wesley.
- Sokolowski, R. (2000). *Introduction to Phenomenology*. Cambridge, UK: Cambridge
 University Press.
- Theman, J. (1983). *Uppfattningar av politisk makt*. Göteborgs universitet, ACTA,
 Göteborg.
- Tranströmer, T. (1979). *Klanger och spår (1966)*. I *Dikter 1954-1978*.
 Stockholm: Bonniers.
- Uljens, M. (1989). *Fenomenografi - forskning om uppfattningar*. Lund:
 Studentlitteratur.
- Wallén, G. (1993). *Vetenskapsteori och forskningsmetodik*. Lund: Studentlitteratur.
- Warren, C. A. B. (2002). Qualitative Interviewing. I J. F. Gubrium & J. A.
 Holstein (Red.), *Handbook of Interview Research: Context & Method*
 (s. 83-100): Thousand Oaks: SAGE Publ.
- Vetenskapsrådet*. (2006). *Forskningsetiska principer*. Avläst 2006-02-15, från:
<http://www.vr.se/publikationer>
- von Wright, G. H. (1986). *Vetenskapen och förnuftet - Ett försök till orientering*.
 Stockholm: Bonnier Fakta Förlag AB.

REFERENSIER

- Winograd, T., & Flores, F. (1988). *Understanding Computers and Cognition*. New Jersey, USA: Ablex Publishing Corporation.
- Yinger, R. J. (1986). Examining thought in action: A theoretical and Methodological critique of research on interactive teaching. *Teaching & Teacher Education*, Vol 2. Nr 3, (s.263-282).

Programmeringens intentionala objekt

I dagens postmoderna samhälle är väsentliga delar i vårt arbete relaterade till datorer i någon form. Robotiseringen inom industrin utgör ett exempel på hur det mänskliga arbetet förskjutits från det konkreta fysiska till att bestå av programmering och övervakning av produktionsprocesser. Denna utveckling ställer nya krav på både utbildning och rekrytering av arbetskraft.

Föreliggande studie är en del i en kartläggning av hur elever begreppsbygger då de arbetar med programmerbart konstruktionsmaterial i teknikundervisningen. Arbetet med programmerbara robotar i skolan är ett exempel på hur tekniska komponenter integreras med datorstött lärande för att ge eleverna en bild av teknik som fenomen i samhället och som introduktion till datorarbete inom skolan.

Studiens vetenskapsteoretiska utgångspunkt ligger inom livsvärldsfenomenologi med metodisk ansats i fenomenografi. Ett grundläggande begrepp inom fenomenologi, och även för denna studie, är intentionalitet. Syftet med uppsatsen är att beskriva vad elever i skolår 8 och 9 riktar sin uppmärksamhet mot när de programmerar – hur elevernas intentionala objekt visar sig när de beskriver sina program. Ambitionen är att resultatet ska ligga till grund för en vidare diskussion kring vilka betydelsefulla faktorer som framträder i elevens begreppsbyggnad av programmering.