Karin Stolpe arbetar som universitetslektor på Institutionen för beteendevetenskap och lärande vid Linköpings universitet. Där forskar hon bland annat inom naturvetenskapernas och teknikens didaktik, med ett särskilt intresse för vilken roll konceptuella metaforer spelar i undervisning. Hon är också intresserad av lärares forskningsanvändning.

Andreas Larsson är postdoktor i teknikens didaktik på Institutionen för beteendevetenskap och lärande vid Linköpings universitet. Hans forskningsintresse rör konceptuella metaforer i undervisning och hans avhandling handlar om programmeringsundervisning. Han är också föreståndare för Nationellt centrum för naturvetenskapernas och teknikens didaktik (NATDID).

KARIN STOLPE
Linköpings universitet, Sverige
karin.stolpe@liu.se

ANDREAS LARSSON
Linköpings universitet, Sverige
andreas.b.larsson@liu.se

”Vi går in och tittar i mappen”

– En lärares språkliga strukturering av rumslighet i programmeringsundervisning

Abstract

Spatial abilities are one of many predictors for students’ learning of programming. This exploratory study aims to investigate how a teacher structure spatiality by means of verbs that indicate motion. Data consists of transcribed talk from YouTube videos that the teacher recorded for teaching purposes. The three most common verbs, surf, drive and go, were chosen for systematic identification of metaphors in scenes. The results indicate that “köra” (‘drive’) has divergent meanings in the context of web server programming, compared to their most basic meaning. However, since “drive” has its specific meaning in the context of programming, the metaphor has been conventionalised. For the verbs “surf” and “go” the metaphoric use indicate a motion into a confound area. The motion has the direction “in” or “out”, which gives some indications of spatiality. Overall, the clues on the spatiality of programming from the teachers’ linguistic utterances are sparce. Implications for education are discussed.

INLEDNING

Den mest grundläggande betydelsen av ordet ”programmera” är att ”analysera ett problem och omforma det till en arbetsinstruktion för en dator” (SO, 2021). För att kunna formulera sådana instruktioner, det vill säga lösa ett programmeringsproblem, behöver man behärska ett programmeringsspråk och man behöver också ha förståelse för en uppsättning programmeringskoncept. Ofta blandas också engelska begrepp och instruktioner med svenska, vilket kan leda till ytterligare utmaningar för eleverna (Becker, 2019).

Spatial förmåga, det vill säga förmågan att tänka i rumsliga relationer och förändringar, har föreslagits vara en prediktor för hur väl elever lyckas lära sig programmering (t.ex. Bockmon et al., 2020; Jones & Burnett, 2008). I de studier som undersöker på vilket sätt elevers spatiala förmåga påverkar deras programmeringskunskaper får eleverna göra standardiserade tester som ofta baseras på olika två- och tredimensionella figurer som ska roteras eller jämföras ur olika vinklar. Typiskt jämförs sedan elevernas resultat på testet med deras betyg i programmering (Jones & Burnett, 2008). Genom att ägna tid åt att låta eleverna träna sin spatiala förmåga genom att tolka olika typer av figurer, verkar eleverna prestera bättre på programmeringsuppgifter jämfört med de elever som bara får träna på programmeringsuppgifterna (Cooper et al., 2015). Samtidigt menar Parkinson och Cutts (2019) att det fortfarande är relativt outforskat om detta samband gäller och varför det i så fall är så. Ett av problemen är att definitionen av spatial förmåga skiljer sig åt mellan olika studier, vilket gör att det blir svårt att dra kumulativa slutsatser. I den här studien väljer vi därför att närma oss spatialitet från ett annat håll, nämligen via språket. Vi utgår från teorier om att de fenomen en individ erfar genom sina kroppsliga upplevelser, kommer påverka hur hens språk är uppbyggt (Lakoff & Johnson, 1980) och vilka ord, uttryck och metaforer hen kommer att använda (Gibbs, 2019). Spatialitet är grundläggande för oss människor i hur vi erfar vår omvärld och hur vi orienterar oss i tid och rum. Ett exempel är när läraren i vår studie säger: ”vi skickar upp filerna på servern”. Här ger ordet ”upp” en rumslighet åt uttrycket. Det är just dessa språkliga konstruktioner som vi är intresserade av i den här studien. Eftersom rumsligheten finns inbyggd i språket kan vi, genom att ta del av hur en lärare talar om programmering, få en indikation på hur läraren relaterar olika begrepp och fenomen till varandra. En lärare kan genom hur hen talar om programmering erbjuda en grund för hur eleverna förstår programmeringens spatialitet.

Denna explorativa studie syftar till att bidra med kunskap om hur en programmeringslärare språkligt strukturerar rumslighet när han undervisar i kursen Webbserverprogrammering i gymnasieskolans teknikprogram. För att studera detta utgår vi från Lakoff och Johnsons (1980) metaforteori. På ett generellt plan tenderar människan att koppla abstrakta koncept till konkreta erfarenheter. Det här sättet att prata om abstrakta saker i termer av kroppsliga erfarenheter benämns här som metaforiskt språk (se Lakoff & Johnson, 1980).

PROGRAMMERING I SKOLAN

Programmering har kommit att bli ett innehåll i många länders skolor under de senaste åren. Därmed har också forskning om programmeringsundervisning kommit att bli allt vanligare. Några av dessa studier rör implementering av olika undervisningsprojekt och effekten därav (Cetin, 2015; Xinogalos, 2016). I ett nordiskt perspektiv fokuserar flera av studierna på elevers lärande av programmering i klassrummet (Anderhag et al., 2021; Cederqvist, 2020, 2022; Kuittinen & Sajaniemi, 2004; Sajaniemi & Kuittinen, 2004). Både lära sig att programmera och att undervisa om och i programmering är svårt (t.ex. Grover et al., 2019; Grover et al., 2015; Ma et al., 2011). Några av de största svårigheterna återfinns på en konceptuell nivå, det vill säga att förstå och hantera programmeringskoncept som variabel, loop och rekursiv funktion. Särskilt svåra att hantera är de koncept som både kräver kunskaper på detaljnivå och kunskaper om hur ett program fungerar på ett större plan (Lahtinen et al., 2005). Lahtinen et al. (2005) menar att det egentligen inte handlar om svårigheten att förstå koncepten *per se,* utan snarare om att förstå hur man ska använda dem. Utifrån sin studie argumenterar forskarna för att undervisning med fokus på praktiska och konkreta moment kan gynna eleverna att möta och överkomma sina utmaningar. Även Cederqvist (2021) pekar på vikten av att till exempel koppla elevernas programmeringsuppgifter till vardagen och till det konkreta för att göra det begripligt. Att kunna förstå olika delar i ett program, eller hur ett program styr olika funktioner i en teknisk artefakt, är ett sätt att göra programmeringen mer konkret. Därmed framstår kopplingen mellan rumslighet och programmeringskoncept.

Lärares intentioner i relation till programmering har också studerats. Vinnervik visar att lärare i grundskolan främst ser programmering som ett medel för att utforska tekniska system vid konstruk-

tionsarbete (Vinnervik, 2021, 2022). Även Rolandsson (2015) menar att programmeringslärare många gånger fokuserar mer på det praktiska görandet snarare än på att grundlägga koncepten.

Flera av studierna ovan indikerar att mycket av programmeringsundervisningen verkar handla om att eleverna ska introduceras till programmering som en aktivitet. De studier som behandlar lärare handlar om lärarnas egna kunskaper om, och intentioner med, programmeringsundervisning. Däremot identifierar vi få studier som undersöker hur själva undervisningen går till och på vilket sätt lärare pratar om programmering. Det finns också förhållandevis få studier som behandlar programmeringsundervisning på gymnasienivå.

METAFORTEORI

Ordet ”metafor” betyder i sin mest grundläggande form ”[ett] uttryck som används om något som liknar det som uttrycket egentligen står för” (SO, 2021). Begreppet är mångfacetterat och används ofta liktydigt med exempelvis analogier, metonymier eller andra lingvistiska konstruktioner. I denna studie lutar vi oss dock mot Lakoff och Johnsons metaforteori (1980), där metaforer är de språkliga konstruktioner som kopplar samman kroppsliga upplevelser med abstrakta fenomen. Det som skiljer deras definition av metaforer från rena språkliga konstruktioner är att dessa formas omedvetet i interaktion med vår omvärld. Lakoff och Johnson (1980) argumenterar för att de koncept som vi har strukturerar det vi upplever, hur vi uppfattar världen och hur vi relaterar till andra människor. Dessa koncept är vi normalt sett inte själva medvetna om, men de speglas i vårt språk. Lakoff och Johnson menar därför att språket fungerar som en viktig källa för att studera hur dessa konceptuella system är konstituerade. Med andra ord speglar vårt språk våra kognitiva processer. På motsvarande sätt kan språk som vi tar del av väcka metaforisk förståelse. I den här studien vill vi därför se hur metaforer strukturerar en lärares tal om programmering. Det blir intressant eftersom det som en lärare säger kommer väcka konceptuella associationer hos de elever som lyssnar till läraren. Vi kan därför få en större förståelse för vilka potentiella tolkningserbjudanden som en lärare bidrar med.

Metaforer inom programmeringsundervisning

Programmering är fullt av uttryck där kopplingar kan göras mellan abstrakta koncept och konkreta erfarenheter. När datorernas grafiska gränssnitt skapades användes ett fysiskt kontor som en liknelse för att användarna skulle känna igen sig (Smith et al., 1982). Därför användes mappar, filer och papperskorgar för funktioner i datorn som skulle underlätta interaktionen mellan människa och dator när textkommandona försvann.

Ett centralt koncept är ”data”, som i undervisningssammanhang talas om i relation till fysiska objekt, till exempel att data kan tas emot (Larsson & Stolpe, 2022). Genom att uttrycka det så här erbjuds åhöraren att tolka data som något konkret som man kan hålla i. Ett annat exempel är att prata om att man ”hoppar mellan kodrader”, något som kan förstås i relation till att förflytta sin kropp från en position till en annan (Colburn & Shute, 2008). Genom en kombination av metaforer i tal och gester gör lärare abstrakta fenomen som listor och rekursiva funktioner konkreta för eleverna (Solomon et al., 2020). Även i läromedlen förekommer olika typer av metaforer, som till exempel ”att bygga datorstrukturer” eller ”flytta objekt” (Larsson, 2022). Studenter använder också metaforer för att beskriva programmeringskoncept (Manches et al., 2020). Även datorprocesser beskrivs metaforiskt som rörelse längs en väg. Alla dessa studier visar att programmering är ett område där metaforer gör det möjligt att prata om det abstrakta i termer av konkreta erfarenheter. På så sätt hjälper metaforerna till att språkligt strukturera ett innehåll.

I utbildningssammanhang spelar metaforerna en stor roll när det handlar om att göra experternas språk tillgängligt för nybörjare (Cameron, 2008). Studier visar exempelvis att metaforer är vanligare när lärare talar till elever eller när en läkare talar till en patient (Cameron, 2008). Fanny et al. (2020) har utgått från Lakoff och Johnsons teori om konceptuella metaforer för att analysera yngre elevers interaktion med robotar. Därigenom har forskarna identifierat tre olika roller som metaforerna

fyller: 1) metaforer som hjälper förståelsen, 2) metaforer som gör det abstrakta konkret, och 3) metaforer som fungerar som slagord. I den första kategorin använde lärarna metaforer för att förklara hur datorns olika delar fungerar. I den andra kategorin användes metaforerna för att göra de abstrakta koncepten konkreta och greppbara för eleverna. I den tredje kategorin användes programmeringen som ett sätt att göra robotarna mer levande och mänskliga.

Samtidigt finns det ord som används i vardagen som har fått en ny betydelse i datorsammanhang. De flesta av oss är medvetna om vad en fil, en mapp, ett fönster och en meny är, både i vardagssammanhang, och när det handlar om datorer (Colburn & Shute, 2008). I många avseenden skulle det därför vara missvisande att benämna dessa begrepp som metaforer. Snarare är de att betrakta som exempel på metaforer som över tid har blivit till konventionaliserade tekniska begrepp (Hidalgo□Céspedes et al., 2018). Det är svårt att säga exakt hur en sådan process går till, men Cameron (2008) poängterar att både processen och resultatet av en konventionalisering ser olika ut i olika språkliga diskurser. Således kan det som är en konceptuell metafor för några vara en konventionaliserad metafor för andra (Müller, 2009).

Orienterande och ontologiska metaforer

Lakoff och Johnson (1980) delar in metaforer i tre olika typer: 1) orienterande, 2) ontologiska och 3) strukturella metaforer. Dessa metaforer etablerar mönster för hur vi förstår saker och hur vi resonerar (Johnson, 1987; Lakoff, 2008). Orienterande metaforer strukturerar hela system av koncept i relation till varandra. De flesta av dessa har att göra med spatial orientering: upp-ner, in-ut, framför-bakom. Därför är denna typ av metaforer av särskilt intresse i vår studie. Lakoff och Johnson menar att spatial orientering härrör från att vi har de kroppar vi har och hur dessa fungerar i den fysiska miljö som vi lever i. Vi kan befinna oss inne eller ute, framför eller bakom. Det är genom dessa erfarenheter vi förstår dessa begrepp.

Ontologiska metaforer innebär att vi ser händelser, aktiviteter, känslor och idéer, som objekt eller substanser. En välkänd ontologisk metafor är container-metaforen (Lakoff & Johnson, 1980), vilken bygger på idén om att fysiska objekt kan förstås i relation till en container med en insida och en utsida som avgränsas av en yta. Vi kan förstå även abstrakta fenomen i termer av container-metaforen, som när vi sparar en fil i en mapp. Genom språket väcks associationer till att mappen fungerar som en container som vi kan lägga något – i det här fallet filen – inuti. På så sätt strukturerar metaforen vårt sätt att förstå relationen mellan mapp och fil, som bygger på våra erfarenheter av att saker kan befinna sig i eller utanför ett föremål.

Redan från det att vi är riktigt små, börjar dessa kognitiva strukturer etableras som tidsrumsliga relationer, och vi lär oss dem i första hand genom våra sensomotoriska upplevelser av världen. Inom kognitiv lingvistik brukar man i den här typen av sammanhang tala om scheman (eng. image schema) som strukturerar våra kognitiva processer och etablerar mönster för hur vi förstår saker och hur vi resonerar (Johnson, 1987; Lakoff, 2008). På svenska kan termen ”föreställningsscheman” som översättning för det engelska begreppet ”image schema” användas (Boström, 2018). Vi väljer här i stället att kalla dem för ”scheman som strukturerar erfarenheter”, vilket ligger i linje med hur Boström (2018) beskriver föreställningsscheman. Vår motivering till att inte använda Boströms begrepp är att ordet ”föreställning” ger signalen att det är ett schema som vi medvetet kan tänka ut. I stället vill vi poängtera att vi redan från det att vi är riktigt små, börjar etablera dessa schematiska strukturer som tidsrumsliga relationer, och vi lär oss dem i första hand genom våra sensomotoriska upplevelser av världen, men vi är ofta inte medvetna om dem (Johnson, 1987).

Ett schema för att strukturera rörelse har en central position i vår studie, då vi utgår från att ett schema om rörelse med dessa element kan hjälpa oss att strukturera rumslighet. Vi utgår från ett schema där en rörelse beskrivs som en rumslig förflyttning från position (A) till en annan position (B) där det rörliga objektet befinner sig vid position (A) vid tiden (T_1) och position (B) vid tiden (T_2) (Filipović & Ibarretxe-Antuñano, 2015). På så vis kan förståelse för tidens ”hastighet” analyseras. Scheman för att förstå rörelse har sin grund i våra erfarenheter av att se ett objekt förflyttas från en plats till en an-

nan (Johnson, 2007; Lakoff & Johnson, 1999). Den här typen av rörelse består av fyra olika element: *Figure*, *Ground*, *Path* och *Motion* (Talmy, 2000). Figure är ett objekt som antingen rör sig eller är fungerar som en referenspunkt för något som rör sig (Moore, 2017; Pavlenko & Volynsky, 2015). Generellt sett kan Figure bestå av mer eller mindre vad som helst så länge den får sin status (exempelvis position) i relation till Ground. Ground fungerar som en referenspunkt mot vilken Figure rör sig. Path motsvaras av den imaginära väg som Figure följer i relation till Ground och Motion "refers to the presence per se of motion or locatedness of the event" (Talmy, 2000, p. 25).

METOD

Datamaterial

Datamaterialet i den här studien har extraherats från 18 YouTube-filmer som har spelats in av en lärare i undervisningssyfte.[1] Vid tiden för studiens genomförande fanns filmerna publikt publicerade på YouTube. De skapades i samband med att distansundervisning periodvis kom att tillämpas i gymnasieskolan i Sverige under pandemiåren 2020-2021. Målgruppen för filmerna är lärarens egna elever som läser kursen Webbserverprogrammering på gymnasieskolans teknikprogram i Sverige (Skolverket, u.å.). Läraren har lämnat samtycke till att filmerna används för denna studie (Vetenskapsrådet, 2017).

Totalt utgör dessa filmer 182 minuter videodata (medelvärde 10 minuter per film). Filmerna utgörs av en skärminspelning av lärarens egen datorskärm. Det gör att man ser vad han gör på skärmen, när han klickar på ikoner, skriver in kommandon, växlar mellan olika fönster och så vidare. Samtidigt har läraren spelat in sin egen röst och beskriver hela tiden det han gör. Han förklarar för åhöraren varför han gör det han gör och hur det hänger ihop. Ibland blir det inte heller som han tänkt sig, och då förklarar han sin felsökningsprocess och vad det var som gick fel. Det är lärarens tal i filmerna som utgör data i den här studien. Samtliga videofilmer har transkriberats ordagrant och består av totalt 24 957 ord (1 386 ord per film i medel).

Databearbetning

Vid en första genomläsning av de transkriberade videofilmerna noterades att läraren använde språket på ett sätt som kan skapa en rumslighet åt det han pratade om. Han sa exempelvis: "Då går vi in i postcontroller", "då ska vi faktiskt hoppa tillbaka här till vår startsida". Vi la märke till att lärarens uttalanden går att tolka som att han själv gick omkring mellan olika filstrukturer, mappar och fönster i datorn. På så sätt byggde rörelserna också upp en rumslighet där rörelse (verb) tillsammans med en riktning (preposition) ger indikationer om var olika datorelement befinner sig i relation till varandra som i exemplet: "vi vill inte hålla på och skicka upp och ner dem på servern". Med den här utgångspunkten ville vi mer systematiskt studera på vilket sätt ett rörelseschema strukturerar rumslighet när läraren pratar om webbserverprogrammering.

Data bearbetades i tre steg för att reducera materialet. I det första steget klistrade vi in transkriptionerna på webbsidan *LIX räknare* (lix.se, LIX betyder Läsbarhetsindex) för att ta reda på vilka ord som var mest frekvent förekommande. Utifrån frekvenslistan identifierade vi verb som skulle kunna indikera någon form av mänsklig rörelse. De tre mest frekvent förekommande verben som skulle kunna relateras till rörelse valdes ut för vidare analys: gå, köra och surfa (tabell 1). Som en jämförelse kan nämnas att det mest frekventa verbet var "ha" (har, ha och hade) och det förekom sammanlagt 442 gånger, men eftersom detta verb, och flera andra som exempelvis "göra", "säga" och "vara" inte syftar på en rörelse har dessa inte analyserats vidare.

I det andra steget noterades den grundläggande betydelsen för vardera av de tre utvalda verben enligt Svensk Ordbok (tabell 1). I Svensk Ordbok ges ofta flera olika definitioner av ett ord. Den första definitionen är den som oftast kan härledas till våra egna kroppsliga erfarenheter, och det är den

[1] YouTube-filmerna finns inte längre publikt tillgängliga.

definitionen som vi här benämner som den mest grundläggande. Om man exempelvis tittar på ordet ”gå” finns hela 11 olika definitioner, men den som uppges först är den som vi använder här.

Tabell 1. De tre mest frekventa verb, och de ordformer som förkommer i datamaterialet, samt den grundläggande betydelsen för dessa verb enligt Svensk Ordbok (SO, 2021).

Verb	Verbformer i data-materialet	Frekvens (totalt)	Grundläggande betydelse (SO)
Gå	Gå Går Gick Gått	114	”förflytta sig genom att växelvis flytta den ena forten fram-för den andra, som då hålls kvar mot underlaget vanligen med underförstådd riktning framåt; om människa el. djur”
Köra	Köra Kör Körs Köras	46	”(ofta med partikel som anger rörelseriktning, t.ex. **av, fram, in, ner, upp, ut**) (driva och) styra fordon el. ekipage, så att det rör sig i önskad riktning” (fetstil i originalet)
Surfa	Surfa Surfar	22	”åka stående på en bräda som bärs fram på vågtoppar eller dras av motorbåt”

I det tredje steget användes programvaran MAXQDA Plus 2020 (ver. 20.3.0) för att genomföra sökningar på de utvalda verben och deras olika ordformer i transkriptionerna. Bara uttalanden som handlade om innehållet, webbserver-programmering kodades. Excerpt från transkriptionerna sammanställdes med hjälp av MAXQDA och exporterades till en Excel-fil. Excerpten utgjordes av kortare meningar, till exempel ”Då ska vi **gå** upp igen och titta på dokumentationen” eller ”så **går** man vidare till index”.

Identifiering av metaforer

Det finns idag ett flertal etablerade metoder (se t.ex. Pragglejaz Group, 2007; Steen et al., 2010) för att systematiskt identifiera konceptuella metaforer. Gemensamt för dessa är att de analyserar språket ord för ord snarare än koncept för koncept (Johansson Falck & Okonski, 2022). Detta görs ofta induktivt för att identifiera eventuella mönster i en större textmassa. Själva analyserna utgår från tanken om att olika ord har olika betydelser i olika sammanhang och att en del av dessa betydelser är mer grundläggande än andra. Genom att jämföra ett ords betydelse i ett visst sammanhang med den mest grundläggande betydelsen, alltså den som oftast har koppling till våra egna kroppsliga erfarenheter, får vi en bild av graden av metaforicitet.

I den här studien är vi dock intresserade av hur olika koncept eller *scener* (se nedan) används snarare än enskilda ord. Vi har därför valt att använda den systematiska metoden PIMS: Procedure for Identifying Metaphorical Scenes (Johansson Falck & Okonski, 2022; Johansson Falck & Okonski, accepterad). En central skillnad mellan PIMS och andra metoder för att identifiera metaforer (Pragglejaz Group, 2007; Steen et al., 2010) är att med PIMS riktas fokus mot en hel scen eller ett koncept, snarare än mot betydelsen av enskilda ord (Johansson Falck & Okonski, 2022). Rörelse studeras alltså i relation till ett schema bestående av flera element, snarare än utifrån ett enskilt verb. Sammantaget utgör elementen i schemat en *scen*.

För att identifiera och avgränsa de scener som vi är intresserade av i den här studien använde vi Talmys (2000) schema för att strukturera rörelse: Figure-Ground-Path-Motion (Pavlenko & Volynsky,

2015). Vårt nästa steg blev således att identifiera de fyra elementen i våra scener. I tabell 2, rad 1, framgår att det går att identifiera tre element. Figure är det objekt som rör sig, det vill sig ”vi”. Path utgörs av den väg vilken objektet rör sig för att komma fram till slutpunkten. I exemplet utgörs path av ordet ”upp”. Motion beskriver den typ av rörelse som äger rum. I det här fallet beskrivs rörelsen av ordet ”gå”. I detta exempel saknas dock Ground, det vill säga den referenspunkt mot vilken Figure rör sig. Därmed är denna scen inte komplett utifrån Talmys schema.

I det andra exemplet (tabell 2, rad 2) utgörs på motsvarande sätt Figure av ordet ”man” och Motion av ”går”. Här finns det en riktning i rörelsen, som utgörs av ordet ”till” och ”index” är den referenspunkt som ”man” rör sig ”till”.

Tabell 2. Två scener från ordet ”gå” som har delats upp i elementen: Figure, Ground, Path och Motion.

Scen	Figure	Ground	Path	Motion
Då ska vi gå upp igen och titta på dokumentationen	vi		upp	gå
så går man vidare till index	man	index	till	går

När scenerna har etablerats är nästa steg i PIMS att etablera och förstå kontexten (Johansson Falck & Okonski, 2022). I den här studien handlar det om att läraren undervisar i webbserverprogrammering. Detta blir avgörande för att kunna identifiera om en scen är metaforisk eller inte. Nästa steg blir därför att avgöra om denna scen *enbart* kan förstås direkt, eller om det finns element som kan förstås i termer av andra (och typiskt mer grundläggande) erfarenheter (Johansson Falck & Okonski, 2022). I det första exemplet skulle scenen kunna förstås bokstavligt. Om vi utgår från ordet ”gå” innebär den grundläggande betydelsen av ordet att vi förflyttar oss med hjälp av att växelvis flytta våra fötter framför varandra (tabell 1). Vi går uppåt, exempelvis för en trappa, för att titta på ett dokument. Men så fort vi väger in att det här sägs i ett sammanhang av webbserverprogrammering får orden en helt annan innebörd, som ligger långt ifrån den bokstavliga, grundläggande betydelsen. Eftersom scenen är inkomplett kan vi dock inte mer precist avgöra vart ”vi” ska ”gå”, annat än att det är i riktning ”upp”. Det kan också förekomma fall där en scen kan förstås direkt, och samtidigt i termer av något annat och kan då sägas vara tvetydig (ambigous) (Johansson Falck & Okonski, 2022), vilket kan uppstå när vi helt enkelt inte har tillräckligt med information. Det kan handla om att en mening inte är fullständig därför att element tas för givna av talaren, eller för att läraren gör en hänvisning med hjälp av muspekaren på skärmen, som vi inte ser i talet.

ANALYS

Analysen för vardera av de tre verben: ”surfa”, ”köra” och ”gå” presenteras nedan. Verben presenteras i ordning från minst frekvent till mest frekvent förekommande i materialet. Verben ska endast ses som utgångspunkt för att hitta och analysera scener som beskriver rörelse.

Surfa

Ordet ”surfa” har kodats 15 gånger i materialet. I exemplet i tabell 3, liksom i samtliga övriga fall där verbet surfa förekommer används verbet på ett sätt som är skiljt från den grundläggande betydelsen och är därmed att betrakta som metaforiskt (jämför tabell 1). Scener där ”surfa” kombineras med ”till” (rad 1, tabell 3) förekommer 11 gånger i materialet. ”Till” har enligt SO (2021) den grundläggande betydelsen ”med (riktning mot och) slutpunkten i el. på el. vid el. hos; i fråga om rörelse el. (tänkt) sträcka med viss bortre gräns”. Ordet ”till” används således av läraren utifrån denna definition och är således inte metaforiskt. Användningen av ”till” indikerar också att det handlar om en rörelse. Det varierar vad ”man” surfar till, men i samtliga fall hänvisar läraren till olika ställen på datorskärmen, som exempelvis ”de få sidor” (tabell 3, rad 1). Det här innebär att ”Ground” i samtliga fall kan

tolkas som en konkret plats eller position på skärmen. Sammantaget blir inte hela scenen metaforisk, även om enskilda ord används metaforiskt. Utifrån ovanstående analys är det rimligt att beskriva ordet ”surfa” utifrån dess konventionaliserade betydelse, som har sin specifika, tekniska betydelse i datorsammanhang. Denna betydelse tas även upp i SO som en specialiserad användning av ordet som hänvisar till att ”besöka sidor på internet eller dylikt” (SO, 2021), ”Surfa” bidrar således inte till mer information om hur rumsligheten struktureras än att man kan ”surfa till” olika positioner. Kontexten spelar en avgörande roll för att tolkningen av scenen.

Men som i exemplet i tabell 3, rad 2, anges en riktning ”in”, som i grundläggande betydelse innebär ”till det inre av något”. I den här scenen struktureras bildmappen som ett avgränsat utrymme som man kan ”surfa direkt in till”. Lärares användning av ordet ”in” ger en metaforicitet till scenen som gör att vi kan föreställa oss att vi faktiskt kan surfa in i mappen. Bildmappen får då också en metaforicitet, och hänvisar inte bara till en ikon på skärmen. Ordet ”surfa” avviker här också från den både den grundläggande, och den mer specificerade betydelsen och gör att filstrukturen kan förstås i termer av att vi kan befinna oss innanför respektive utanför en bildmapp.

Tabell 3. Ordet ”surfa” där scenen har delats upp i elementen: Figure, Ground, Path och Motion.

Scen	Figure	Ground	Path	Motion
man kan surfa direkt in till en bildmapp	man	bildmapp	till	surfa

Köra

Ordet ”köra”, med dess olika ordformer, har kodats 28 gånger i materialet. Ordet används metaforiskt i samtliga scener. I dess grundläggande betydelse av att styra ett fordon kombineras detta ord ofta med ett ord som anger i vilken riktning färden går (tabell 1). Det blir då intressant att notera att i väldigt få scener (3 av 28) kombineras ordet med både Ground och Path. I 25 av fallen är scenen inkomplett och det saknas information om både Ground och Path (tabell 4). Det här betyder att ”köra” inte används som ett ord som benämner rörelse, med en riktning och ett mål. I stället används det för att benämna igångsättandet av en process, vilket syns i exempel som ”vi kör bara vår funktion”, eller ”Då måste vi säga åt den att köra ’Post find’”. Det går således att argumentera för att ordet ”köra” får en avgränsad och specifik betydelse i det sammanhang som utgörs av programmeringskontexten, och som skiljer sig från den mer grundläggande och kroppsligt förankrade erfarenheten av att styra ett fordon i en viss riktning. Istället liknar användningen den definition av ”köra” som också tas upp i SO ”få (maskin eller dylikt) att arbeta på avsett sätt” (SO, 2021). Även om ”köra” utifrån denna argumentation används metaforiskt, ger scenerna inga ledtrådar om rumsligheten i webbserverprogrammering.

Tabell 4. Ordet ”köra” där scenen har delats upp i elementen: Figure, Ground, Path och Motion.

Excerpt	Figure	Ground	Path	Motion
Den här if-satsen ska köras när saker händer	if-satsen			köras

Gå

Ordet ”gå” har kodats 65 gånger i materialet och används alltid metaforiskt, det vill säga skiljt från den grundläggande betydelsen. Eftersom ”gå” används så pass många gånger har vi också kunnat göra en mer finmaskig analys på hur ordet används beroende på vilka ord det kombineras med. Som framgår av tabell 5 är de flesta scener där ”in” och ”ut” används som Path metaforiska. Däremot har de flesta scener där ”till” eller ”tillbaka” en låg grad av metaforicitet. Det är också relativt hög andel av de sistnämnda som är tvetydiga. En förklaring till detta är att när läraren pratar i termer av ”in” och ”ut”

används samtliga element (Figure, Ground, Path, Motion) metaforiskt. Som i exemplet nedan (tabell 5, rad 1), är vendor-mappen i kontexten av webbserverprogrammering något annat än en ”enklare, pärmliknande anordning (av plast, papper eller dylikt) för förvaring” (SO, 2021). Vendor-mappen hänvisar här till en logisk enhet i datorns filsystem som grupperar filer. Men i kombination med orden ”vi kan gå in” i mappen, kan vendor-mappen förstås i termer av något som vi fysiskt kan gå in i. Vendor-mappen kan därmed tolkas i termer av något annat. Eftersom alla element i den här scenen används metaforiskt är helheten metaforisk. Analysen visar att detta är typiskt när ”gå” kombineras med ”in” eller ”ut” (tabell 5).

När ordet ”till” används uppstår en annan situation. Här syftar ordet ”till” oftast på en konkret struktur på skärmen, vanligen en mapp eller en fil. Även om ordet ”gå” fortfarande används metaforiskt, finns en direkt koppling mellan det som syns på skärmen (mappen) och ordet ”mapp”. Det innebär att bara vissa element av scenen är metaforiska och scenen kan därmed förstås användas med låg metaforicitet utifrån kontexten webbserverprogrammering.

Tabell 5. Antalet scener som har identifierats som möjliga att tolka som hög metaforicitet, respektive låg metaforicitet och tvetydiga där ordet ”går” har varit utgångspunkt, i kontexten webbserverprogrammering.

Exempel på scen	Path	Hög metaforicitet	Låg metaforicitet	Tvetydigt
Vi kan gå in och titta i vendor-mappen	in	11	3	3
Sen ska den gå ut till ”epic” [mappen]	ut	4	0	0
Så får man gå till rätt fönster	till	2	10	5
så går vi tillbaka till index	tillbaka	0	7	6

DISKUSSION

De tre ord som var de vanligaste verben som indikerade någon form av rörelse: ”gå”, ”köra” och ”surfa” användes på olika sätt av läraren. Analysen visar att de tre orden fungerar på olika sätt när de används i den specifika kontexten när en lärare talar om webbserverprogrammering.

Ordet ”köra” som i dess mest grundläggande betydelse har att göra med att styra ett fordon i en viss riktning, har en annan betydelse i sammanhanget av webbserverprogrammering, nämligen att exekvera ett program eller en funktion i en dator. Ordet ”köra” i sig har i det här sammanhanget blivit så pass etablerat att vi sällan tänker på att det uppfyller kriteriet för att användas metaforiskt. Många ord som används i datorsammanhang har på det här sättet sitt ursprung i vardagliga begrepp, men har fått en alltmer konventionaliserad betydelse (Colburn & Shute, 2008). I och med att ”köra” inte passar in i den schematiska strukturen för rörelse kan vi dra slutsatsen att ”köra” inte används överhuvudtaget för att beskriva en rörelse i det här sammanhanget. ”Köra” skulle i dessa scener också kunna beskrivas som en död metafor (Müller, 2009). Därmed kan ordet inte ge oss mer information om hur läraren pratar om rumslighet.

När det gäller de scener där ordet ”surfa” används är hela den schematiska strukturen för rörelse uppfylld. Däremot uppfyller inte samtliga element kriteriet för att tolkas som metaforiska. Ordet ”surfa” används metaforiskt, det vill säga skiljt från den grundläggande betydelsen. Men eftersom det handlar om att surfa ”till” en viss plats på skärmen betraktas den som konkret. Det finns alltså en direkt koppling mellan ordet som man surfar till, till exempel ”mapp”, och en position som syns på skärmen. Kontexten webbserverprogrammering ger ”surfa” en ny, konventionaliserad betydelse. Således kan inte heller ”surfa” ge oss en vidare inblick i rumsligheten hos programmering. Däremot finns det fall där ordet ”surfa” kombineras med orden ”in till”. Då får ”surfa” innebörden av att vi kan ta oss från ett yttre till ett inre, vilket gör att inte bara den schematiska strukturen för rörelse aktiveras. Även det

som brukar benämnas som container-schemat (Lakoff & Johnson, 1980) visar hur läraren använder rumslighet för att förstå mer abstrakta resonemang. Ordet ”surfa” ger alltså i vissa scener en insyn i rumslighet.

Ordet ”gå” liknar ordet ”surfa”. När ”gå” kombineras med ordet ”till”, uppstår samma typ av mönster som med ”surfa”. Det som läraren pratar om att man ”går till” är en konkret position på skärmen. Därför blir den schematiska strukturen för rörelse enbart delvis metaforisk, vilket gör att vi förstår scenen med en låg grad av metaforicitet.

Det finns också tillfällen när ”går” kombineras med ”in”. På samma sätt som för ordet ”surfa” aktiveras då container-schemat (Lakoff & Johnson, 1980). Vi kan inte gå in i en filstruktur, men ändå pratar läraren på det sättet. Det ger oss en tydligare bild av programmeringens spatialitet, nämligen att vissa saker kan tänkas befinna sig inne i andra. Eftersom vi använder mentala scheman (t.ex. rörelse-schemat eller container-schemat) för att förstå och göra abstrakta beskrivningar begripliga, kan den här typen av verbala konstruktioner som läraren gör, hjälpa elever att strukturera sin förståelse av programmeringens spatialitet.

Det som också synliggörs genom analysen är att läraren använder sig av relativt få konceptuella metaforer som relaterar till den schematiska strukturen för rörelse. Det resultatet stärks också utav det faktum att läraren i väldigt låg grad använder sig av verb som har koppling till kroppslig rörelse, jämfört med andra verb.

Metoddiskussion

Den här studien är att betrakta som en explorativ fallstudie och bidrar med kunskap om hur språket kring webbserverprogrammering kan strukturera rumslighet. För att göra studiens resultat trovärdigt och tillförlitligt är systematik och transparens centralt. Vi har därför använt en strukturerad metod (Johansson Falck & Okonski, 2022) för att identifiera metaforer. Dessutom har vi vinnlagt oss om att beskriva varje steg i analysen noggrant. Lärarens användning av metaforer säger något om hur han kognitivt strukturerar programmering, men vi kan aldrig veta hur han tänker. Däremot kan vi genom vår analys uttala oss om den språkliga strukturen som vi kan ta del av, och grunden för den finns i lärarens kognitiva strukturer. På det sättet kan vi som forskare inta samma position som eleverna, som mottagare av ett budskap (för ett vidare resonemang, se Cienki, 2016).

Den typ av metaforanalys som används i den här artikeln är starkt kopplad till de individer som genomför analysen. Vi måste alltså vara medvetna om att vår egen förförståelse och våra egna kognitiva scheman, påverkar vår möjlighet att tolka lärarens verbala uttalanden. Här spelar också språket roll. En mottagare med ett annat modersmål än lärarens, skulle kunna påverka förståelsen för metaforer. Vi som forskare har samma modersmål som den lärare som ingår i studien, därför kan vi inte uttala oss om vilka möjligheter en elev med ett annat modersmål har att förstå ett budskap som metaforiskt. Här kan vi exempelvis göra kopplingen till ”köra” som betyder två väldigt olika saker beroende på om det tolkas som den grundläggande betydelsen att styra ett fordon i en viss riktning, eller den betydelse som är kopplad till att exekvera en funktion. Olika elever med olika bakgrund och förförståelse kommer således att uppfatta lärares tal på olika sätt (se även Becker, 2019). Detta är extra viktigt att vara medveten om i undervisningssammanhang.

Den här studien är avgränsad till att endast titta på en lärares verbala yttranden när han undervisar programmering framför en datorskärm. Filmen är förinspelad och läraren har ingen interaktion med några elever medan han pratar. Tidigare studier har även visat att metaforer förekommer inom programmeringsundervisning, men inte heller i dessa studier har fokus legat på interaktionen i klassrummet (Larsson & Stolpe, 2022; Solomon et al., 2020). Vidare forskning måste därför göras för att kunna svara på om de mönster som vi har sett här återupprepas även i en klassrumssituation. Interaktion med eleverna, samt genom att studera hur läraren interagerar med skärmen när han rör muspekaren i olika riktningar, skulle kunna bidra med ytterligare intressanta aspekter av hur konceptuella metaforer strukturerar talet om programmering.

SLUTSATSER OCH IMPLIKATIONER

Genom att den här studien är explorativ till sin karaktär prövar vi om användningen av metaforteori kan hjälpa till att förstå vilken rumslighet en programmeringslärare strukturerar språkligt. Tidigare forskning har visat sig vara en nyckel för elever för att förstå programmering (Bockmon et al., 2020; Jones & Burnett, 2008). Däremot har inte tidigare studier fokuserat på språkets betydelse för att skapa denna rumslighet. Vi är därmed i den här studien intresserade av att undersöka på vilket sätt en lärares språkliga konstruktioner gav ledtrådar om programmeringens rumslighet. Genom att använda rörelseschemat ”Figure, Ground, Motion, Path”, visade analysen att verb som ”surfa” och ”köra” främst används som ett sätt att navigera mellan olika strukturer på skärmen respektive att starta processer i datorn. Därmed kan konstateras att användningen av vissa verb är konventionaliserad, något som med andra ord kan uttryckas som att metaforerna är döda (Müller, 2009).

En slutsats som vi kan dra är att alla verb som vi i vardaglig bemärkelse kopplar samman med rörelse, inte fungerar som indikation på rörelse i en programmeringskontext. Därmed kan man tala om att det språk som läraren pratar i relation till programmeringen blir ett språk som skiljer sig från elevernas vardagsspråk. Det skulle kunna potentiellt skapa missförstånd hos eleverna, kanske framför allt för de elever som inte talar svenska som modersmål.

Däremot ser vi att metaforiskt språk blir synligt i relation till webbserverprogrammering i vår studie, vilket bekräftas i tidigare forskning (Colburn & Shute, 2008; Hidalgo-Céspedes et al., 2018; Larsson, 2022; Larsson & Stolpe, 2022; Manches et al., 2020). Det kan alltså konstateras att det för eleverna inte bara handlar om att lära sig ett programmeringsspråk, som till exempel Python eller C++, utan också ett språk för att prata om programmering (se även Becker, 2019).

Genom vårt sätt att analysera data kan vi också dra slutsatsen att scener som innehåller språkliga konstruktioner som surfa in till” (tabell 3), ”gå in” eller ”gå ut” (tabell 5), kan tolkas utifrån ett container-schema. Läraren pratar exempelvis om att ”vi går in i en mapp”. De här resultaten ligger i linje med att data konceptualiseras som ett objekt som kan ligga ”i en mapp” (Larsson & Stolpe, 2022). Programmeringens spatialitet är således bara delvis framträdande genom språket. Den här delen av analysen skulle kunna stärkas än mer om man också tar hänsyn till videofilmen. Om videon då visar att muspekaren rör sig över en ikon som indikerar en mapp, så skulle evidensen stärkas för att läraren faktiskt pratar om en position på skärmen. I den här studien har vi dock valt att avgränsa oss till verbala uttalanden. Vidare forskning kan svara på vilken relation det som händer på skärmen har till det som uttalas verbalt.

REFERENSER

Anderhag, P., Björn, M., Fahrman, B., Lundholm-Bergström, A., Weiland, M., & Wållberg, T. (2021). Kod som teknisk lösning: en studie om grundskoleelevers uppfattningar av ändamålsenlighet i deras spontana programspråk. *Nordic Studies in Science Education*, *17*(1), 113-129. https://doi.org/10.5617/nordina.7020

Becker, B. A. (2019). Parlez-vous Java? Bonjour La Monde!= Hello World: Barriers to programming language acquisition for non-native English speakers. Proceedings of the 30th Annual Workshop of the Psychology of Programming Interest Group (PPIG'19), 28-30 August 2019, Newcastle University, UK.

Bockmon, R., Cooper, S., Gratch, J., Zhang, J., & Dorodchi, M. (2020). Can Students' Spatial Skills Predict Their Programming Abilities? Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, June 15-19, 2020, Trondheim, Norway.

Boström, P. (2018). *”Det här är ju dött tåg liksom…”: en studie av metaforer för ROMANTISK KÄRLEK i talad svenska* [Doktorsavhandling, Umeå universitet].

Cameron, L. (2008). Metaphor and talk. In R. W. Gibbs Jr (Ed.), *The Cambridge handbook of metaphor and thought* (pp. 197-211). Cambridge University Press.

Cederqvist, A.-M. (2020). Pupils' ways of understanding programmed technological solutions when analysing structure and function. *Education and Information Technologies*, *25*(2), 1039-1065. https://doi.org/https://doi.org/10.1007/s10639-019-10006-4

Cederqvist, A.-M. (2021). Designing and coding with BBC micro:bit to solve a real-world task–a challenging movement between contexts. *Education and Information Technologies*, 355-381. https://doi.org/https://doi.org/10.1007/s10639-021-10865-w

Cederqvist, A.-M. (2022). An exploratory study of technological knowledge when pupils are designing a programmed technological solution using BBC Micro:bit. *International Journal of Technology and Design Education*, *32*, 355-381. https://doi.org/10.1007/s10798-020-09618-6

Cetin, I. (2015). Students' understanding of loops and nested loops in computer programming: An APOS theory perspective. *Canadian Journal of Science, Mathematics and Technology Education*, *15*(2), 155-170. https://doi.org/https://doi.org/10.1080/14926156.2015.1014075

Colburn, T. R., & Shute, G. M. (2008). Metaphor in computer science. *Journal of applied logic*, *6*(4), 526-533. https://doi.org/https://doi.org/10.1016/j.jal.2008.09.005

Cooper, S., Wang, K., Israni, M., & Sorby, S. (2015). Spatial skills training in introductory computing. Proceedings of the eleventh annual international conference on international computing education research, Omaha, Nebraska, USA.

Fanny, B., Julie, H., & Anne-Sophie, C. (2020). Developing a critical robot literacy for young people from conceptual metaphors analysis. 2020 IEEE Frontiers in Education Conference (FIE), 21-24 October, 2020, Uppsala, Sweden.

Filipović, L., & Ibarretxe-Antuñano, I. (2015). Motion. In *Handbook of cognitive linguistics* (pp. 527-546). De Gruyter Mouton.

Gibbs, R. W. (2019). Metaphor as Dynamical-Ecological Performance. *Metaphor and Symbol*, *34*(1), 33. https://doi.org/https://doi.org/10.1080/10926488.2019.1591713

Grover, S., Jackiw, N., & Lundh, P. (2019). Concepts before coding: non-programming interactives to advance learning of introductory programming concepts in middle school. *Computer Science Education*, *29*(2-3), 106-135. https://doi.org/10.1080/08993408.2019.1568955

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, *25*(2), 199-237. https://doi.org/10.1080/08993408.2015.1033142

Hidalgo-Céspedes, J., Marín-Raventós, G., Lara□Villagrán, V., & Villalobos□Fernández, L. (2018). Effects of oral metaphors and allegories on programming problem solving. *Computer Applications in Engineering Education*, *26*(4), 852-871. https://doi.org/https://doi.org/10.1002/cae.21927

Johansson Falck, M., & Okonski, L. (2022). Procedure for Identifying Metaphorical Scenes (pims): A Cognitive Linguistics Approach to Bridge Theory and Practice. *Cognitive Semantics*, *8*(2), 294-322. https://doi.org/https://doi.org/10.1163/23526416-bja10031

Johansson Falck, M., & Okonski, L. (accepterad). Procedure for identifying metaphorical scenes (PIMS): The case of spatial and abstract relations.

Johnson, M. (1987). *The body in the mind: The bodily basis of meaning, imagination and reason Chicago*. The Univ. of Chicago Press.

Johnson, M. (2007). *The meaning of the body: Aesthetics of human understanding*. University of Chicago Press. https://doi.org/10.7208/chicago/9780226026992.001.0001

Jones, S., & Burnett, G. (2008). Spatial ability and learning to program. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*.

Kuittinen, M., & Sajaniemi, J. (2004). Teaching roles of variables in elementary programming courses. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds, UK.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *Acm sigcse bulletin*, *37*(3), 14-18. https://doi.org/https://doi.org/10.1145/1151954.1067453

Lakoff, G. (2008). *Women, fire, and dangerous things: What categories reveal about the mind*. University of Chicago press.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. University of Chicago Press.

Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: the embodied mind and its challenge to Western thought*. Basic Books.

Larsson, A. (2022). Reading the Code Between the Lines –Exploring the Structure of metaphors in educational programming resources. *Nordic Studies in Science Education*, *18*(3), 305-322. https://doi.org/https://doi.org/10.5617/nordina.8774

Larsson, A., & Stolpe, K. (2022). Hands on Programming: Teachers' use of Metaphors in Gesture and Speech make Abstract Concepts Tangible. *International Journal of Technology and Design Education*. https://doi.org/https://doi.org/10.1007/s10798-022-09755-0

Ma, L., Ferguson, J., Roper, M., & Wood, M. (2011). Investigating and improving the models of programming concepts held by novice programmers. *Computer Science Education*, *21*(1), 57-80. https://doi.org/10.1080/08993408.2011.554722

Manches, A., McKenna, P. E., Rajendran, G., & Robertson, J. (2020). Identifying embodied metaphors for computing education. *Computers in Human Behavior*, *105*, 105859. https://doi.org/https://doi.org/10.1016/j.chb.2018.12.037

Moore, K. E. (2017). Elaborating time in space: The structure and function of space–Motion metaphors of time. *Language and Cognition*, *9*(2), 191-253. https://doi.org/https://doi.org/10.1017/langcog.2016.6

Müller, C. (2009). *Metaphors dead and alive, sleeping and waking: A dynamic view*. University of Chicago Press.

Parkinson, J., & Cutts, Q. (2019). Investigating the relationship between spatial skills and computer science. *ACM Inroads*, *10*(1), 64-73.

Pavlenko, A., & Volynsky, M. (2015). Motion encoding in Russian and English: Moving beyond Talmy's typology. *The Modern Language Journal*, *99*(S1), 32-48. https://doi.org/https://doi.org/10.1111/modl.12177

Pragglejaz Group. (2007). MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, *22*(1), 1-39. https://doi.org/10.1080/10926480709336752

Rolandsson, L. (2015). *Programmed or Not: A study about programming teachers' beliefs and intentions in relation to curriculum* [Doktorsavhandling, KTH Royal Institute of Technology].

Sajaniemi, J., & Kuittinen, M. (2004). Visualizing roles of variables in program animation. *Information Visualization*, *3*(3), 137-153. https://doi.org/https://doi.org/10.1057/palgrave.ivs.9500075

Skolverket. (u.å.). Webbserverprogrammering [Kursplan]. https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen/amne?url=1530314731%2Fsyllabuscw%2Fjsp%2Fsubject.htm%3FsubjectCode%3DWES%26tos%3Dgy&sv.url=12.5dfee44715d35a5cdfa92a3

Smith, D. C., Irby, C., Kimball, R., & Harslem, E. (1982, June 7-10, 1982). The Star user interface: An overview. National Computer Conference, Houston, Texas, USA.

SO. (2021). *Svensk ordbok*. svenska.se

Solomon, A., Bae, M., DiSalvo, B., & Guzdial, M. (2020). Embodied Representations in Computing Education: How Gesture, Embodied Language, and Tool Use Support Teaching Recursion. The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020, Volume 4, Nashville, Tennessee, USA.

Steen, G., Dorst, A., G., Hermann, J. B., Kaal, A., Krennmayr, T., & Pasma, T. (2010). *A method for linguistic metaphor identification. from MIP to MIPVU*. John Benjamins Publishing.

Talmy, L. (2000). *Toward a cognitive semantics, volume II: Typology and process in concept structure*. MIT Press.

Vetenskapsrådet. (2017). *God forskningssed*. Vetenskapsrådet.

Vinnervik, P. (2021). *När lärare formar ett nytt ämnesinnehåll: Intentioner, förutsättningar och utmaningar med att införa programmering i skolan* [Doktorsavhandling, Umeå universitet].

Vinnervik, P. (2022). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*, *32*(1), 213-242. https://doi.org/10.1007/s10798-020-09602-0

Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Education and Information Technologies*, *21*(3), 559-588. https://doi.org/ https://doi.org/10.1007/s10639-014-9341-9