

Lärares föreställningar om kunskap och lärande i datorteknik

Lennart Rolandsson, fil lic., lärare i matematik, datateknik och programmering vid gymnasieskolan

Ett rundabordssamtal¹ hölls tidigare i år vid Utbildningsdepartementet, kring IT i skola och utbildning. Samtalet påminde mycket om det som utspelade sig under 1980-talet då datorteknik ställde nya krav på lärares kompetens inom IT. Att uppgradera lärares ämneskunskaper i IT är av stor vikt, men det räcker inte. Den didaktiska förmågan i relation till ämnesinnehållet är minst lika viktig. Svensk didaktisk forskning har mer sällan uppmärksammat problemet om undervisning i IT. Vi har därför begränsade kunskaper i området mellan IT, lärares föreställningar om kunskap och elevers lärande.



Lennart Rolandsson

UNDER EN STOR DEL av 1980-talet var undervisning i datorteknik synonymt med programmering. Från flera auktoriteter höjdes kritiska röster om lämpligheten av programmering i skolan; innehållet var alltför tekniskt till sin natur. Sverige förde en utbildningspolitik som syftade till att så många som möjligt skulle erbjudas datorundervisning (Datadelegationen, 1985)², vilket innebar att merparten av eleverna erbjöds undervisning i programmering under sista året i grundskolan och/eller första året på gymnasiet. Utöver denna allmänbildning erbjöds även undervisning i datateknik vid

naturvetenskapliga linjen vid gymnasiet.

I en statlig utredning som föregick utformandet av läroplanen, Lpf 94 står att ansvaret för skolans utveckling ligger på ”de pedagogiskt professionella” och att behovet av reflektion kring lärares föreställningar är nödvändigt för att nå målen.

”När nu skolväsendet är under omvandling till ett mer målstyrt system, där de pedagogiskt professionella har ansvar för utvecklingen av verksamheten i skolan, blir diskussionen om de kunskaps- och inlärningsteoretiska grunderna för denna verksamhet än viktigare. En

¹ På Regeringskansliet den 21/2 2011. Rundabordssamtalet ingick i en serie inom Digitala rådet.

² Utbildningsradion drev en kurs riktad till allmänheten i 10 delar. En av dessa handlade om programmering.

professionell lärarverksamhet innebär ställningstaganden till såväl kunskap, lärande och undervisning som till hur dessa omsätts i lärarverksamheten. En sådan verksamhet kan inte styras genom föreskrifter om dess utformning. Istället är det samspelet mellan lärarnas teoretiska föreställningar och verksamhetens praktiska utformning som bör göras till grund för en alltmer reflekterad praktik.” (Läroplanskommittén, 1994, s.28–29)

Den svenska gymnasieskolans IT-undervisning har sällan undersökts inom didaktisk forskning. Det finns några arbeten (Nissen, Riis, & Hyltén, 1991; Riis, 1991) som undersöker hur datorer påverkar skolan och lärares förmåga att förändra sin praktik (Jedeskog, 2000). Följande licentiatuppsats undersöker lärares föreställningar i dag och historiskt och därmed existerande begränsningar för undervisningen, med förhoppningen att kunna bidra till kunskapsutvecklingen på området.

Forskningsfokus

Undervisningens villkor om IT och datorteknik har förändrats sedan datorerna gjorde sin skoldebut på 1970-talet. Datortekniken har genomgått en snabb utveckling och politiska beslut har skapat nya förutsättningar för undervisningen inom IT. Dessutom utvecklas nya programmeringsmiljöer och programmeringsspråk i snabb takt, vilket ställer krav på att läraren kontinuerligt uppdaterar sig med nytt ämnesinnehåll.

Licentiatuppsatsen beskriver vad som hänt i Sverige under nästan 30 år genom att undersöka vilka didaktiska överväganden som programmeringslärare uttrycker i sin praktik. Med stöd av

forskning om lärares epistemiska föreställningar om *kunskap och lärande* (Hofer & Pintrich, 2002; Khine, 2008) har jag i mitt licentiatarbete valt att undersöka om och i så fall hur synen på lärande i programmering har förändrats.

I min licentiatuppsats *Programming Teachers' Epistemic Beliefs and Curriculum Intentions* (Rolandsson, 2012) har jag valt att avgränsa undersökningen till följande aspekter:

1. Ett historiskt perspektiv med frågeställningen:

Hur gick det till då svenska kursplaner i datorteknik och programmering skrevs fram under 1970-, 1980- och 1990-talen?

2. Ett nutidsperspektiv där svenska gymnasielärares föreställningar kring ämnets och undervisningens förutsättningar är i fokus, med frågeställningen:

Vilka epistemiska föreställningar finns bland programmeringslärare av en utbildning som riktar sig till alla?

Tidigare forskning

Ämnesdidaktiska studier inom programmering på gymnasienivå med ett historiskt förklaringsvärde är få till antalet (Micheuz, 2005; Rolandsson, 2011; Syslo & Kwiatkowska, 2005). Om man dessutom söker förstå undervisningen från lärarens perspektiv blir de än färre (Rolandsson, 2009). Ett antal undersökningar där man sätter programmeringslärares undervisningskunskaper i relation till ämnets innehåll (Ragonis & Hazzan, 2008; Saeli, Perrenet m fl, 2011; Woollard, 2005) har också genomförts³.

³ Ofta inspirerade av PCK-modellen (Shulman, 1992).

Metod

Historiskt perspektiv

När det gäller den historiska kartläggningen av datorrämnets införande har flera typer av källor använts. För en översikt av kommunikationen mellan lärare och Skolöverstyrelsen (SÖ) har tidskrifter från 1970-, 1980- och 1990-talen använts. Historiska källdokument från Riksarkivet och olika Stadsarkiv har undersökts. Ett tiotal personer som var engagerade i kursplaneutvecklingen för data-teknik/datakunskap har intervjuats. Även källor från Riksdagens bibliotek, som belyser politiska intentioner och datorer i skolan har använts.

Data av relevans för kursplaneutveckling, från Riksarkiv och olika stadsarkiv, skannades och sammanställdes i digitala volymer för att granskas i efterhand; interaktionen mellan olika arbetsgrupper inom SÖ och mellan SÖ och olika skolor/lärare studerades speciellt.

Nutidsperspektiv

För att nå kunskap om erfarenheter av undervisning och lärande i programmering startades ett lärarnätverk dit en majoritet av alla svenska lärare i programmering inbjöds. Inom ramen för en seminarierie med fyra tillfällen, under 2009 till 2011, diskuterades olika didaktiska teman/frågeställningar, lärares erfarenheter och arbetsmetoder.⁴

I samband med seminarierna tillfrågades lärarna via enkäter om sina erfarenheter och didaktiska överväganden i programmeringsundervisningen. Insamlade data analyserades och kategoriserades i en tvåstegs-process. En preliminär analys visade på två olika synsätt vad gäller elevers förutsättningar

att lära sig programmering; det fanns lärare som ansåg, och de som inte ansåg, att *alla* elever kan lära sig programmering. I nästa steg genomfördes en induktiv dataanalys (Lincoln & Guba, 1985; Miles & Huberman, 1994) för att utveckla underkategorier.

RESULTAT

Programmering i skolan – ett historiskt perspektiv

En försöksverksamhet startade, i slutet av 1970-talet, med datateknisk inriktning vid naturvetenskapliga programmet. Man utgick från en samtida eftergymnasial yrkesutbildning för programmerare. Löften om utveckling av elevers logiska förmågor och problemlösning genom kunskaper i programmering var attraktiva. Inom skolämnet för datateknik undervisades även systemutveckling för att kunskaper inom programmering skulle framställas i ett socialt sammanhang.

Programmering i skolan ifrågasattes tidigt (Clements & Gullo, 1984; Reed & Palumbo, 1991); det var svårt att påvisa en generell utveckling av högre kognitiva förmågor (Palumbo, 1990). Enligt forskning från 1980-talet (Dalbey & Linn, 1985; Linn & Dalbey, 1985; Pea & Kurland, 1984) var undervisningen deklarativ med mycket tekniska detaljer. I Sverige valde man att utveckla ett unikt programmeringsspråk, DPG-PROLOG, och en skoldator, COMPIS (Utbildningsdepartementet, 1988), för svenska förhållanden. Datorutvecklingen rusade fram och kunskaper inom programmeringen ansågs överflödiga och ersattes därför med expertsystem och datorapplikationer. Kunskaper inom programmering återkom dock i nästa läroplansreform.

Elbranschen uttryckte ett behov av kompetens

⁴ Om möjlighet ges finansiellt och resursmässigt kommer seminarierien fortgå. Träffarna genomfördes i samarbete med Teknikum/Stockholms universitet, Uppsala universitet, KTH och Microsoft Sweden AB.

bland elever som läser på elprogrammet. Därför erbjöds kurser inom programmering i och med Lpf 94. Det är första gången som kunskaper i programmering fick eget ämne. Kurserna erbjöds så småningom till alla elever som så önskar. Idag läser cirka 6 000 elever programmering varje år.

Lärarnas föreställningar – ett nutidsperspektiv

Följande karakteristika har framkommit i undersökningen:

- 1) En majoritet av lärarna, utgår ifrån att *inte alla* elever som påbörjar kurserna kommer att lära sig programmering. Avgörande faktorer enligt lärarna är elevers intresse/motivation, sammanhängande tid och elevers logiska/analytiska förmåga.
- 2) Det finns ett uttalat individuellt lärande i klassrummet.
- 3) Undervisningen varierar från att vara programmeringsteknisk (med fokus på teknikaliteter i programmeringsspråket) till att erbjuda eleverna generella programmeringsfärdigheter och kunskaper inom problemlösning.
- 4) Flertalet lärare arbetar med enskild handledning och korta genomgångar.
- 5) En stor majoritet av lärarna använder industrispråk snarare än anpassade utbildningsspråk för programmering.

Slutsatser och diskussion

Studien visar att programmering på gymnasiet oftast är ett individuellt lärande med flera mindre uppgifter, där mycket tid läggs på teknikaliteter inom ett eller flera programmeringsspråk. En

sådan undervisning riskerar att fokusera på deklarativa kunskaper vilket inte stimulerar till utveckling av högre kognitiva förmågor (Palumbo, 1990). Studien visar dessutom att lärarna av idag har höga förväntningar på elevers logiska och analytiska förmågor (även innan kurserna påbörjats). Sammantaget pekar därför avhandlingen på en exkluderande undervisning i programmering.⁵

Kanske är programmeringsundervisning idag överflödigt, utifrån ett samhällsperspektiv. Ämnet gick från att vara en yrkesprofession, blev sedan till allmänbildning, för att sedan introduceras under egen kurs för de intresserade och motiverade. Dock passar det inte alla elevgrupper.

Då samma exkluderande undervisningsmönster återfinns även inom andra skolämnen som till exempel matematik (Ball & Bass, 2000), kan man ana att mötet mellan programmering, eleven och läraren inte bara varit en tillfällighet i datorteknikens utveckling. Det finns ett annat problem i botten.

Förändring av lärares föreställningar är krävande (Burns m fl, 2002), vilket 30 år av mötet mellan datorteknik, elev och lärare visat. Seymour Papert (1980) skapade ett undervisningsspråk, LOGO, som fokuserade mera på logisk förståelse än förståelse av programmeringsspråket själv. I Sverige slog det *inte* igenom.⁶ Idag finns andra alternativa programmeringsspråk för undervisning⁷ men programmeringslärare föredrar tunga industrispråk snarare än språk och utvecklingsmiljöer anpassade för undervisning. Varför föredrar lärare att utforma en undervisning som inte inkluderar en mångfald av elevers olika sätt att lära? Licentiatavhandlingen

⁵ Ett liknande mönster förekommer redan under 1980-talet (Dalbey & Linn, 1985; Linn & Dalbey, 1985).

⁶ Enligt Saarikoski (Saarikoski, 2011) arbetade man i Finland målmedvetet med att införa ett undervisningsspråk, LOGO, i undervisningen för att undvika att fastna i programmeringsspråkens olika teknikaliteter.

⁷ T.ex. BASIC, Pascal, BlueJ, Greenfoot, Alice, Python och Kodu.

pekar på svaret, då den identifierat ett stort behov, som existerat under lång tid. Programmeringslärare behöver mer än ämneskunskaper; minst lika viktigt är att lärare utvecklar en didaktisk repertoar baserad på reflektion kring mötet mellan IT, elever och deras egna föreställningar.

Referenser

- Ball, D. & Bass, H. (2000). Interviewing content and pedagogy. I: J. Boaler (red.), *Multiple perspectives on mathematics teaching and learning*. Westport, CT: Ablex Pub.
- Burns, M., Menchaca, M. & Dimoc, V. (2002). Applying technology to restructuring and learning. *CSCL '02 Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, s. 281.
- Clements, D.H. & Gullo, D.F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76, s. 1051–1058.
- Dalbey, J. & Linn, M.C. (1985). The demands and requirements of computer programming: A literature review. *Journal of Educational Computing Research*, 1(3), s. 253–274.
- Datadelegationen. (1985). *Bred datautbildning: Betänkande*. Stockholm: Liber/Allmänna förlaget.
- Hofer, B.K. & Pintrich, P.R. (red.). (2002). *Personal epistemology: The psychology of beliefs about knowledge and knowing*. Mahwah, N.J.: L. Erlbaum Associates.
- Jedeskog, G. (2000). *Teachers and computers: Teachers' computer usage and the relationship between computers and the role of the teacher*. Pedagogiska institutionen. Uppsala universitet.
- Khine, M.S. (red.). (2008). *Knowing, knowledge and beliefs: Epistemological studies across diverse cultures*. New York: Springer.
- Läroplanskommittén. (1994). *Bildning och kunskap: Särtryck ur läroplanskommitténs betänkande skola för bildning (SOU 1992:94)*. Stockholm: Statens skolverk. Liber distribution.
- Lincoln, Y.S. & Guba, E.G. (1985). *Naturalistic inquiry*. Beverly Hills, California: Sage Publications.
- Linn, M.C. & Dalbey, J. (1985). Cognitive consequences of programming instruction: Instruction, access, and ability. *Educational Psychologist*, 20(4), s. 191–206.
- Micheuz, P. (2005). 20 years of computers and informatics in Austria's secondary academic schools. *ISSEP 2005, LNCS*, 3422, s. 20–31.
- Miles, M.B. & Huberman, A.M. (1994). *Qualitative data analysis: An expanded sourcebook*. Beverly Hills, California: Sage Publications.
- Nissen, J., Riis, U. & Hyltén, B. (1991). *Skolan och datorn: Delrapport 2: Besök våren 1991 vid 23 skolor som bedriver försök med datorn som pedagogiskt hjälpmedel*. Linköpings universitet.
- Palumbo, D.B. (1990). Programming Language/ Problem-solving research: A review of relevant issues. *Review of Educational Research*, 60(1), s. 65–89.
- Papert, S. (1980). *New cultures from new technologies*. s. 230–240.
- Pea, R. D. & Kurland, M. (1984). The cognitive effects of learning computer programming. *New Ideas in Psychology*, 2(2), s. 137–168.
- Ragonis, N. & Hazzan, O. (2008). Disciplinary-pedagogical teacher preparation for pre-service computer science teachers: Rational and implementation. *LNCS*, 5090, s. 253–264.
- Reed, W. & Palumbo, D. (1991). The effect of BASIC programming language instruction on high school students' problem-solving ability and computer anxiety. *Journal of Research on*

- Computing in Education*, 23(3), s. 343–372.
- Riis, U. (1991). *Skolan och datorn: Satsningen datorn som pedagogiskt hjälpmedel 1988–1991*. Linköpings universitet.
- Rolandsson, L. (2009). Teachers' perceptions about learning programming. *Proceedings PATT-22 Conference. Strengthening the Position of Technology Education in the Curriculum*, Delft, The Netherlands.
- Rolandsson, L. (2011). Teacher pioneers in the introduction of computing technology in Swedish upper secondary school. I J. Impagliazzo m fl (red.). *History of Nordic Computing 3*. New York: Springer-Verlag.
- Saarikoski, P. (2011). Computer courses in Finnish schools during 1980–1995. I J. Impagliazzo m fl (red.). *History of Nordic Computing 3*. New York: Springer-Verlag.
- Saeli, M., Perrenet, J., Jochems, W.M.G. & Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, 10(1), s. 73–88.
- Shulman, L. (1992). Ways of seeing, ways of knowing, ways of teaching, ways of learning about teaching. *Journal of Curriculum Studies*, 28, s. 393–396.
- Syslo, M.M. & Kwiatkowska, A.B. (2005). Informatics versus information technology – how much informatics is needed to use information technology – A school perspective. *ISSEP 2005, LNCS*, 3422, s. 178–188.
- Utbildningsdepartementet. (1988). *Försöksverksamhet med Prolog i undervisningen vid gymnasieskolan*. Stockholm: Utbildningsdepartementet.
- Woollard, J. (2005). The implications of the pedagogic metaphor for teacher education in computing. *Technology, Pedagogy and Education*, 14(2), s. 189–204.