

hyväksymispäivä arvosana

arvostelija

## **Alan Kayn ideat ja Smalltalk-kielen synty**

Isto Havu

Helsinki 15.2.2007

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto – Fakultet/Sektion – Faculty/Section		Laitos – Institution – Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä – Författare – Author			
Havu Isto			
Työn nimi – Arbetets titel – Title			
Alan Kayn ideat ja Smalltalk-kielen synty			
Oppiaine – Läroämne – Subject			
Tietojenkäsittelytieteen historia -seminaari			
Työn laji – Arbetets art – Level		Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages
Seminaaritutkielma		15. helmikuuta 2007	11
Tiivistelmä – Referat – Abstract			
<p>Alan C. Kay ryhtyi 1970-luvun alussa johtamaan Xerox PARC:n tutkimuskeskuksessa ryhmää, jonka tavoitteena oli luoda helppokäyttöinen olioperustainen ohjelmointikieli sekä kannettava henkilökohtainen tietokone. Kayn tavoitteet perustuivat hänen kokemuksiinsa tietokoneista 1960-luvulla ja arvioonsa tietokoneiden yleistymisestä. Tutkimusryhmän työn tuloksena syntyi oliokieli nimeltä Smalltalk.</p> <p>ACM Computer Classification System (CSS): K.2 [History of computing]</p>			
Avainsanat – Nyckelord – Keywords			
Smalltalk, Kay, oliokieli, graafinen			
Säilytyspaikka – Förvaringställe – Where deposited			
Muita tietoja – Övriga uppgifter – Additional information			

## Sisältö

1	Johdanto .....	1
2	Kayn ideat luovat unelman.....	1
2.1	Ensikosketus tietokoneisiin ja oliot.....	1
2.2	Lisää laitteistoja ja käyttötapoja.....	2
3	Syntyy Smalltalk - ja paljon muutakin .....	3
3.1	Alku on graafinen.....	3
3.2	Lapsikin osaa ohjelmoida Smalltalkilla?.....	4
3.3	Hallitsevatko aikuiset Smalltalkin?.....	5
3.4	Projekti vastatuulella .....	6
3.5	Smalltalk valmistui – Dynabook ei .....	7
4	Mikä on Smalltalk? .....	8
5	Yhteenveto .....	10
	Lähteet.....	11

## 1 Johdanto

Alan Curtis Kay oli ensi kertaa tietokoneiden kanssa tekemisissä 1960-luvun alussa. Tietokoneet olivat silloin valtavan kokoisia ja ainoastaan alan ammattilaisten lähinnä tieteellisiin tarkoituksiin käyttämiä. 60-luvun loppuun mennessä Kay alkoi uskoa teknisen kehityksen myötä tietokoneesta tulevan vähitellen kotien ja koulujen yleinen laite. Hän itse ryhtyi toteuttamaan tätä visiota Xeroxin tutkimuskeskuksessa tavoitteenaan kannettava tietokone ja järjestelmä, jota jokainen osaisi käyttää ikään ja koulutukseen katsomatta. Syntyi ohjelmointikieli nimeltä Smalltalk sekä lukuisia hyviä ideoita, joita käytämme nykyisissä tietokonejärjestelmissämme päivittäin.

Tekstin sisältö perustuu Kayn kirjoitukseen The Early History of Smalltalk [Kay93] ellei muuta lähdettä mainita.

## 2 Kayn ideat luovat unelman

Kay tutustui tietokoneisiin 1960-luvun alussa, jolloin ne olivat suuria ja vaikeasti käytettäviä. Käyttäessään silloista alan tekniikkaa ja tutustuessaan uusiin tietoteknisiin innovaatioihin, hän sai unohtumattomia ideoita siitä, miten tietokoneen ja sen järjestelmän voisi toteuttaa. Kokemuksien myötä Kaylle syntyi selkeä visio tulevaisuuden tietokoneesta ja sen toteuttamisaikataulusta.

### 2.1 *Ensikosketus tietokoneisiin ja oliot*

Vuonna 1961 Kay oli Yhdysvaltain ilmavoimien palveluksessa ja ensimmäistä kertaa tekemisissä tietokoneiden kanssa. Hän ohjelmoi Burroughs 220 ja 5000 –koneita, jotka vaativat paljon tilaa. Niiden tiedostot ja käyttöjärjestelmät saattoivat olla rakenteeltaan erilaisia jopa kahden samanlaisen ja samassa käytössä olevan koneen välillä. Jotta kokonaisen ohjelman siirto koneelta toiselle olisi onnistunut, oli tiedostot jaettu kolmeen osaan: dataan, dataa käsitteleviin prosedureihin sekä proseduurit käynnistäviin osoittimiin. Tästä ratkaisusta Kay katsoo saaneensa ensimmäisen kerran ajatuksen olioperustaisesta ohjelmointikielestä.

Kay suoritti alemman asteen tutkinnon (Bachelor of Science) Coloradon yliopistossa 1966 pääaineinaan matematiikka ja molekyylibiologia. Hän jatkoi opintojaan Utahin yliopistossa, jossa tietojenkäsittelytieteen koulutusohjelmaa johti Dave Evans. Evansin johdatuksella Kay

tutustui Sketchpadiin, joka oli ensimmäinen vuorovaikutteinen graafinen käyttöympäristö: ohjauskynällä piirrettiin graafisia elementtejä kokonaisuudessaan kolmasosaneliömailin (noin 0,77 neliökilometrin) laajuisella näytöllä zoomattavalla työalalla.

Kay perehtyi myös Simula-kieleen<sup>1</sup>, joka oli laajennettu versio Algol-60:stä. Simulan ja biologian opintojensa pohjalta Kay sai ajatuksen, että tietokoneessakin toiminnot voitaisiin jakaa pienille itsenäisille yksiköille, aivan kuten eliöissä toimintoja suorittavat solut. Sen sijaan, että kieli jakautuisi yksinään heikkoihin osiin, kuten proseduureihin tai tietorakenteisiin, voisi se jakautua olioihin, jotka toimisivat ”pieninä, itsenäisinä tietokoneina”.

## **2.2 Lisää laitteistoja ja käyttötapoja**

Vuonna 1967 Evans esitteli Kayn Edward Cheadlelle, joka työskenteli paikallisen ilmailu- ja avaruusalan yrityksen palveluksessa. Cheadlen ohjauksessa Kay alkoi lopputyönään kehittää korkeamman tason ohjelmointikieltä FLEX-konetta varten. FLEX, joka oli ensimmäisiä yhden käyttäjän tietokoneita, oli resursseiltaan liian pieni Simulalle. Niinpä Kay kehitti konetta varten FLEX-ohjelmointikielen, jossa oli paljon vaikutteita Simulasta. Sen ensimmäinen versio perustui kääntämiseen, mutta Kay kiinnostui myös suoran tulkkauksen mahdollisuudesta. Tämä johti ajatukseen helposti laajennettavasta olioperustaisesta kielestä. Lopulta hän luopui kääntämisestä ja loi version, jossa jokainen olio tulkitisi saamansa viestit syntaksiin perustuen. Työ valmistui vuonna 1969 ja tuloksena oli ensimmäinen graafinen olioperustainen henkilökohtainen tietokone.

Kesällä 1968 Kay piti konferenssiesitelmän FLEX-koneesta ja näki ensimmäisen litteän plasmanäytön Illinoisin yliopistolla. Vain yhden tuuman koosta huolimatta näyttö synnytti Kaylle vision tulevaisuuden tietokoneista. Hän laski Mooren lailla<sup>2</sup>, kuinka pian olisi mahdollista tehdä laskentateholtaan FLEXin veroinen mutta kooltaan niin pieni kone, että sen piisirut sopisivat litteän näytön taakse: aikaisintaan 1970- ja 80-lukujen vaihteessa.

Samana vuonna Kay tutustui järjestelmään nimeltä GRAIL (Graphic Input Language), jolla konstruointiin vuokaavioita. Työssä ei käytetty lainkaan näppäimistöä, vaan järjestelmää ohjattiin graafisesti ohjauskynää käyttäen. GRAIL oli riittävä osoitus Kaylle siitä, että ohjelmointikielen käyttöliittymä voisitoin pelkän ohjauskynän tai hiiren avulla.

<sup>1</sup> Proseduraalisen Simula-simulaatiokielen kehittivät norjalaiset Kristen Nygaard ja Ole-Johan Dahl [NyD66].

<sup>2</sup> Gordon Moore ennusti piisirujen tehon kasvavan tilavuuteen nähden eksponentiaalisesti [Moo65].

Vaikutuksen Kay'hin teki myös Seymour Papertin projekti, jossa tutkittiin lasten kykyjä ohjelmoida. Tarkoitusta varten oli kehitetty LOGO-kieli, jolla lapset osasivat ohjata näytöllä olevan pisteen ja mekaanisen kilpikonnän liikkeitä [Sol78].

Kay alkoi näkemänsä perusteella ajatella, että henkilökohtainen tietokone olisi parhaimmillaan media, jonka käyttö olisi mahdollista iästä ja koulutuksesta riippumatta. Unelmaksi muodostui tietokone, joka ei olisi muistilehtiötä suurempi, jonka järjestelmä olisi helppokäyttöinen kuin GRAIL tai LOGO ja kehittynyt kuin Simula tai FLEX. Kay teki pahvista mallikappaleen ja ujutti sen sisään lyijypainoja arvioidakseen ihannepainon: yksi kilo. Luomus sai nimekseen Dynabook. Järjestelmän joustavuus ja laajennettavuus perustuisi olioihin, jotka Kayn mielessä olivat tarkentuneet syntaktisesti ohjattaviksi tulkeiksi. Jokaisella oliolla olisi virtuaalikoneessa oma syntaksinsa ja muistinsa, joilla ne tulkitsisivat niille lähetettyjä viestejä.

### **3 Synty Smalltalk - ja paljon muutakin**

Smalltalk syntyi Xerox PARCissa (Palo Alto Research Center) Kayn 1971 perustaman Learning Research Groupin työn tuloksena. Xeroxin vuotta aiemmin perustetun tutkimuskeskuksen tavoitteena oli kehittää tulevaisuuden toimistolaitteita ja sovelluksia, mutta Kayn tavoitteena oli luoda kannettava tietokone, jossa olisi graafinen käyttöympäristö ja ohjelmointikieli, jota ”lapsikin osaisi käyttää”. Tätä ajatusta pidettiin vaikeana toteuttaa ja suorastaan utopistisena, mutta Kay palkkasi mukaan vain ideasta innostuneent ihmiset. Dan Ingalls oli Kayn mukaan ryhmän ideoiden käytännön toteuttaja ja toimeenpanija. Adele Goldberg tutki Smalltalkin soveltuvuutta lapsille. Hän johti ryhmän työn loppuun aina Smalltalkin kaupallisen version julkaisemiseen Kayn vetäytyttyä 1980-luvun alussa.

#### **3.1 Alku on graafinen**

Koska Kay uskoi voivansa toteuttaa Dynabookin vasta vuosikymmenen päästä (1980), hän tyytyi aloittamaan kehitystyön pöydällä pidettävästä kookkaammasta koneesta. Siitä hän käytti nimeä Kiddi Komp ja myöhemmin miniCOM. Heti vuonna 1971 syntyi myös ensimmäinen, tosin melko puutteellinen versio ohjelmointikielestä, Smalltalk-71. Kay halusi nimen luonnehtivan kieltä ja sen tavoiteltuja ominaisuuksia: ”ohjelmoinnin tulee olla kuin smalltalkia” ja ”lasten kelpaa ohjelmoida smalltalkilla” [Kay93, sivu 528].

Graafinen käyttöliittymä muodosti heti aluksi käytännön ongelman: näytölle mahtui tekstiä graafisessa tilassa huomattavasti vähemmän kuin tekstiilassa – liian vähän. Jo vuoden 1971 aikana Kay keksi ratkaisuksi päällekkäiset ikkunat, joista aktivoitu näkyisi aina päällimmäisenä. Näin näytölle saatiin moninkertainen määrä tietoa. Ikkunat eivät tuolloin tuntuneet vallankumoukselliselta idealta, vaan ainoastaan kokeilemisen arvoiselta mahdollisuudelta, joiden ansiosta käyttöliittymän kehitys lähti vauhtiin. Ikkunat toteutettiin heti sen jälkeen kun Smalltalk oli saatu lukemaan näppäimistöä ja esittämään merkkijono.

Kirjasinlajeja ja päällekkäisiä ikkunoita kokeiltiin keväällä 1972. Samaan aikaan Xeroxin johdon kanssa alkoi esiintyä erimielisyyttä tutkimusryhmän tarpeista. Kay oli toteuttanut miniCOMin paperilla osista, jotka olivat helposti saatavilla. Xeroxin johto ei kuitenkaan nähnyt henkilökohtaisten tietokoneiden kehittämistä tärkeäksi. Kayn tavoitteena oli monimediaan kykenevä viihdelaitte (playstation), kun Xerox uskoi työaseman (workstation) kehittämisen olevan taloudellisesti järkevintä. Seurauksena Kay ei saanut pyytimiään resursseja miniCOMien rakentamiseen. Syyskuussa 1972 projektiin kielteisesti suhtautunut Xeroxin johtaja oli muissa tehtävissä ja tilaisuus käytettiin hyväksi: Learning Research Group sai käyttöönsä Xeroxin kehittämän Alto-tietokoneen, joka ristittiin Bilboksi. Smalltalkia alettiin ajaa Bilbolla, jota oli tarkoitus käyttää ”väliaikaisena Dynabookina” kunnes tutkimusryhmä saisi resurssit todellisen Dynabookin rakentamiseen.

Smalltalk-72:lla luotiin useita sovelluksia, joita hyödynnettiin kehitystyössä. Yksi näistä oli miniMOUSE-tekstieditori, joka oli Xerox PARCin ensimmäinen WYSIWYG-ohjelma – tekstin ulkoasu oli siis jo näytöllä muokattaessa sama kuin tulostettaessa. Siitä tuli pian käytetyin sovellus koodin kirjoittamiseen. Käyttöliittymän graafisuus ja ikkunat mahdollistivat mediasovellusten tuottamisen. Smalltalkilla toteutettiin esimerkiksi kaksiulotteista animaatiota sekä piirto- ja taittosovellukset, joiden työkalut olivat valittavissa kuvakeriveistä näytöllä.

### **3.2 Lapsikin osaa ohjelmoida Smalltalkilla?**

Smalltalkista oli tarkoitus tehdä ohjelmointikieli, joka olisi nopeasti opittavissa ilman erityistä koulutusta tai aikaisempaa kokemusta ohjelmoinnista. Kielen syntaktisen suppeuden lisäksi yksinkertaisuutta lisäsi se, että kaikki ohjelmointiin tarvittavat työkalut löytyivät hiirellä käytettävistä valikoista.

Oli luonnollista kokeilla tavoitteen toteutumista lapsilla, jotka myös osoittautuivat ihanteellisiksi kielen koekäyttäjiksi [HPW88]. He olivat innokkaita kokeilemaan, koska tunsivat voivansa määrätä konetta. Innokkuutta lisäsi se, että lapset eivät kokeneet tekevänsä harjoituksia tai leikkivänsä, vaan luovansa uusia asioita. Heillä ei ollut aiempaa kokemusta ohjelmoinnista, joten he syntyivät ohjelmoijina Smalltalk-ympäristöön. Lisäksi lasten mielenkiinto säilyi asiaan tuntien ajan. Kokeet alkoivat kesällä 1973, jolloin kieli oli kehittynyt riittävälle tasolle. Koska tutkimusryhmässä kenelläkään ei ollut aikaisempaa kokemusta lasten kanssa työskentelystä, vahvistukseksi hankittiin muiden muassa Adele Goldberg Stanfordin yliopistosta.

Smalltalkiin oli jo edellisenä vuonna kehitetty luokka, joka oli olioperustainen versio LOGO-kilpikonnasta. Ensimmäiset kokeet lasten kanssa tehtiin kilpikonnilla samaan tapaan kuin Seymour Papert oli tehnyt [Sol78]. Tulokset olivat vastaavia: lapset osasivat ohjelmoida liikkeitä, mutta pelkkä kilpikongan ohjaaminen ei ollut kovin hedelmällistä.

Goldberg kokeili seuraavaksi ohjelmapätkän esittämistä lapsille, joiden tuli arvata lopputulos. Ohjelmissa piirrettiin ja liikutettiin näytöllä neliöitä ja muita kuvioita. Kuviot olivat olioita, joilla saattoi olla ihmisten nimet. Näin oliokäsite konkretisoitui lapsille. Lähestymistavan kautta lapset oppivat pian ohjelmoimaan kuvioiden liikkeitä ja muodostamaan useampia yhdistelemällä monimutkaisia kuvioita näytölle. Jotkut oppivat kielen niin hyvin, että he ohjelmoivat kokonaisia piirtosovelluksia, joissa piirtotyökalujen valikoima koostui neliöistä, ympyröistä ja muista kuvioista.

Lapset, jotka täyttivät odotukset luodessaan Smalltalkilla todellisia sovelluksia, veivät tutkimusryhmän huomion kokonaisuudelta, joka ei ollut yhtä lupaava. Kokeisiin valikoituneet lapset olivat Palo Alton alueen koulujen oppilaita. Monien vanhemmat olivat töissä Xeroxilla tai muissa teknologiayrityksissä, joten heidän perhetaustaansa voi pitää otollisena ajatellen tietotekniikan omaksumista. Silti heistäkin vain pieni osa pystyi neliönpyörittelyä syvällisempään Smalltalkin käyttöön.

### **3.3 Hallitsevatko aikuiset Smalltalkin?**

Keväällä 1974 Smalltalkia testattiin ohjelmointiin perehtymättömillä aikuisilla. He oppivat kielen syntaksin ja käyttöliittymän hallinnan lapsia nopeammin, mutta todellisten ongelmien ratkaisu, kuten osoitemuistion toteuttaminen Smalltalkilla, tuottivat heille ratkaisun lyhydestä huolimatta ylitsepääsemättömiä vaikeuksia.



Ongelmaksi havaittiin lasten ja aikuisten kohdalla sama asia. Niin kauan kuin tehtävä oli ratkaistavissa yhtä luokkakirjaston työkalua käyttäen, ratkaisu oli ilmeinen. Mutta jos yhden tehtävän ratkaisuun tarvittiin useampia työkaluja, ei ratkaisu ollut enää helppo, vaikka kaikki työkalut olivat poimittavissa valikoista. Esimerkiksi osoitemuistion vain kahden sivun pituisesta lähdekoodista Kay löysi 17 epäilmeistä ratkaisua. Oikeiden perättäisten työkalujen yhdistelmän löytäminen luokkakirjaston satojen mahdollisten joukosta osoittautui niihin perehtymättömälle käyttäjälle lähes mahdottomaksi tehtäväksi. Myös periytyminen, jonka oletettiin helpottavan sovellusten rakentamista valmiiksi tehtyjen luokkien avulla, osoittautui niin aloittelijoille kuin kokeneille ohjelmoijille vaikeasti omaksuttavaksi ominaisuudeksi.

### **3.4 Projekti vastatulessa**

Koko tutkimusryhmän työtä ohjasi Kayn näkemys tietoteknisen alan tulevaisuudesta. Hän sai toistuvasti perustella projektiaan Xeroxille ja osui tulevaisuuden ennusteissaan nykyhetken näkökulmasta hämmästyttävän oikeaan. Tämä käy ilmi esimerkiksi hänen 1970-luvun puolivälissä Xeroxille lähettämästään muistiosta [Kay93, sivu 551] (oma käänös):

1990-luvulla maailmassa on miljoonia henkilökohtaisia tietokoneita. Ne ovat muistilehtiön kokoisia, niissä on litteät korkearesoluutioiset heijastavat näytöt, ne painavat alle kymmenen kiloa ja niissä on [silloiseen] Altoon verrattuna 10-20-kertainen laskenta- ja muistikapasiteetti. [...]

Vaikka Dynabookissa on merkittävä tallennuskapasiteetti ja se tekee laskennan pääosin itse, on se valtaosan ajasta kytkettynä maailmanlaajuisiin informaatio-sovelluksiin, jotka tekevät mahdolliseksi kommunikoinnin muiden kanssa ideoiden, datan, työasioiden kuten myös päivittäisten kuulumisten vaihtamiseksi.

Jatkuvasta edistyksestä huolimatta vuoden 1975 lopussa tutkimusryhmän jäsenet kokivat, että sille asetetut päämäärät olivat jääneet toteutumatta. Ajatus lasten ohjelmointikielestä ei enää ollut lähtökohta kehitystyössä eikä Alto ominaisuuksineen korvannut tarvetta rakentaa Dynabook. Tammikuussa 1976 ryhmä kokoontui keskustelemaan tilanteesta ja tavoitteista. Päällimmäisenä nousi esiin tyytymättömyys Smalltalkin tehokkuuteen: sovellusten suorittaminen oli liian hidasta eikä suurten ongelmien ohjelmallinen ratkaiseminen ollut mahdollista. Kielen olioperustaisuus ei ollut vastoin odotuksia ollut ratkaisu tehottomuuteen kuten ei myöskään se ominaisuus, joka tekisi kielestä erityisen helposti omaksuttavan. Keskustelun perusteella päädyttiin työnjakoon, jossa Kay otti vastuulleen pienen kannettavan

tietokoneen, NoteTakerin suunnittelun. Siitä oli tarkoitus tehdä jälleen Dynabookin edeltäjä, sen hetken saatavilla olevilla osilla. Dan Ingalls alkoi johtaa Smalltalk-76:n kehittämistä.

Elokuussa 1976 tuli jälleen takaisku. Xerox PARCissa oli jo 500 Altoa käytössä, lähiverkolla toisiinsa ja lasertulostimeen yhdistettynä. Uusin versio, Alto III oli jo niin pieni, että se sopi työpöydälle. Kaikesta huolimatta Xerox ei uskonut Alton tai henkilökohtaisten tietokoneiden mahdollisuuksiin markkinoilla ja päätti jättää Alton kaupallistamatta. Päätös oli Kayn ryhmälle merkki siitä, ettei Xeroxin strategia antaisi eväitä Smalltalkin tai etenkin Dynabookin kehittämiseen.

Marraskuussa 1976 Ingalls sai valmiiksi Smalltalk-76:n, jonka lähdekoodi ja luokkakirjasto olivat kirjoitettu kokonaan uudelleen – tuloksena oli huomattavasti aikaisempia parempi Smalltalk. Sen suorituskyky ei enää ollut rajoituksena ohjelmoitavien sovellusten koolle. Luokkakirjaston 50 luokkaa koostuivat 180 sivullisesta lähdekoodia ja sisälsivät kattavat toiminnot käyttäjärjestelmää, tiedostoja, tulostamista, lähiverkkoa, ikkunoita, editoreita, grafiikkaa ja piirto-sovelluksia varten. Smalltalk-76 sisälsi tutkimusryhmän vuosien varrella jalostamat ideat toimivassa kokonaisuudessa ja kaikki myöhemmät versiot ovat läheisesti muistuttaneet sitä.

### **3.5 Smalltalk valmistui – Dynabook ei**

Tammikuussa 1978 Smalltalk-76 joutui tulikokeeseen. Kymmenen Xeroxin johtajaa oli kutsuttu tutkimuskeskukseen tutustumaan kehitystyöhön ja myös Kayn ryhmän oli määrä esitellä heille projektinsa saavutuksia. Aikaisemmista kokeiluista oppineena Smalltalkia esiteltiin sovelluksen avulla. Lapsi- ja aikuistestaajien kanssa oli käytetty Simpula-mallia, jolla oli mahdollista simuloida esimerkiksi huvipuiston tai tuotantolinjan toimintaa. Simpula esikuvana tehtiin pikaisesti kehittyneempi Smalltalk SimKit, jossa oli graafinen käyttöliittymä. SimKit esitti animoidulla grafiikalla sille syötetyn tuotantolinjan toimintaa ja salli helposti muuttaa siihen vaikuttavia tekijöitä. Kolmen tunnin esittely oli onnistunut: yhdeksän kymmenestä johtajasta onnistui itse ohjelmoimaan haluamansa tuotantotilanteen. Yksi simuloinneista, joka perustui todellisiin tekijöihin Xeroxin omistaman yhtiön tietokoneiden tuotantolinjasta, paljasti suuria puutteita työntekijöiden sijoittelussa.

Vuoden 1978 aikana Kay sai valmiiksi NoteTakerin, mutta ei ollut tyytyväinen lopputulokseen. Ulkonäöltään kone muistutti miniCOMia ja kooltaan suurta salkkua. Tehokkuudessa ei ollut moittimista: kahdella Intelin 8086-prosessorilla Smalltalk-78:n

tulkkaus sujui lähes kaksinkertaisella nopeudella verrattuna Altoon. NoteTaker oli kannettava tietokone ja toimi myös akuilla, mutta painonsa ja kiinteän näyttönsä asennon takia sitä oli pakko käyttää työpöydällä. Kilon painoisen Dynabookin toteutuminen vaikutti jälleen kaukaiselta haaveelta. Kay oli eniten harmissaan siitä, että teknisesti NoteTaker olisi ollut toteutettavissa jo vuosia aikaisemmin, jos Xerox olisi ollut valmis käyttämään siihen hallussaan olevaa teknologiaa ja varoja.

Vuonna 1979 tutkimusryhmä sai vieraakseen Applen Lisa-projektin työntekijöitä. Projektilla oli päämääränä luoda henkilökohtainen tietokone graafisella käyttöliittymällä. Projektin avainhenkilö Steve Jobs innostui nähtyään mihin Smalltalk kykeni ja yritti ostaa Xeroxilta tutkimusryhmän kehittämää teknologiaa. Siitä huolimatta, että Xerox oli yksi Applen riskirahoittajista, ei Jobs saanut käyttöönsä Kayn ryhmän luomuksia. Kaiken lisäksi Xerox päätti pian tämän jälkeen lopettaa NoteTakerin kehittämisen.

Koska tutkimusryhmän jäsenten alkuperäinen tavoite – luoda Dynabook – oli näin tullut mahdottomaksi toteuttaa, ei yhteistä päämäärää enää ollut. Vuonna 1980 osa siirtyi Steve Jobsin palkkaamina Lisa-projektiin Applelle. Kayn suurimmat kannustimet, Dynabook ja lasten ohjelmointikieli eivät olleet enää toteutettavissa, joten hän otti vuoden vapaata, teki lyhyen työrupeaman Atarille ja sen jälkeen projekteja Applelle. Goldberg ja Ingalls keskittyivät Smalltalkin hiomiseen julkaisukelpoiseksi. Smalltalk-78:sta syntyi hyvin pienin muutoksin virallisesti vuonna 1983 julkaistu Smalltalk-80.

## 4 Mikä on Smalltalk?

Kahdella sanalla kuvattuna Smalltalk on olioperustainen ohjelmointikieli. Laajemmin tarkasteltuna se on enemmän kuin pelkkä kieli: se on integroitu järjestelmä, joka sisältää kaiken tarvittavan ohjelmiston kehittämiseen, testaamiseen ja suorittamiseen. Smalltalk on kolmen komponentin – oliokielen, luokkakirjaston ja kehitysympäristön – yhdistelmä.

Kuten jo nimi Smalltalk vihjaa, kieli on syntaksiltaan moniin muihin ohjelmointikieliin nähden suppea. Itse kielellä on mahdollista määritellä muuttujat, liittää muuttujat olioihin ja lähettää oliolle viestejä. Kaikki toiminnot – laskutoimitukset, silmukat, syötteet ja tulosteet – toteutetaan olioiden toisilleen lähettäminä viesteinä, joihin saavat vastauksetkin ovat oliota. Myös luvut, merkkijonot ja luokat ovat oliota. Monia ohjelmointikieliä luonnehditaan

olioperustaisiksi, mutta Smalltalkissa todella kaikki perustuu olioihin. Sen yksinkertainen syntaksi ei sisällä perustoimintoja, joita ohjelmointikieliltä yleensä odotetaan. Ne sisältyvät satojen luokkien luokkakirjastoon. Esimerkiksi luvut ja loogiset arvot ovat kirjastossa valmiiksi määriteltyjä olioita. Aritmeettiset toimitukset toteutetaan lähettämällä viestejä luvuille ja loogiset lausekkeet ratkaistaan lähettämällä totuusarvo oliolle [Lew95].

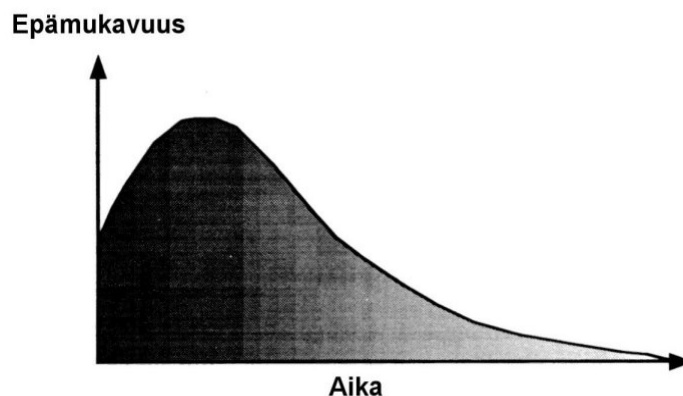
### Smalltalkin lause

```
13 > (3+4) ifTrue: [Transcript show: 'Suurempi'].      (1)
```

voidaan tulkita seuraavasti. 3 on olio, jolle lähetetään viestinä +4. Olion 3 vastaus viestiin on 7. Oliolle 13 lähetetään viestinä > 7 ja se antaa vastauksen true. Oliolle true lähetetään viesti ifTrue: ja parametrit [Transcript show: 'Suurempi']. Koska arvo true kuuluu luokkaan True, suoritetaan parametrit, eli lähetetään oliolle Transcript viesti show: ja parametrit 'Suurempi'. Olio Transcript tulostaa ikkunaan tekstin Suurempi. Todellisuudessa Smalltalk tulkitsee lauseen (1) vasemmalta oikealle, rekursiivisesti. Siten edellä esitetyn lauseen (1) tulkinnan ensimmäisessä vaiheessa 13 on olio, jolle lähetetään viesti > ja parametreinä lauseen koko loppu.

Koska Smalltalk on semanttisesti ja syntaktisesti yksinkertainen ja hyvin pelkistetty, se on kielenä nopeasti opittavissa. Silti laajan, täydennettävän ja muokattavan luokkakirjaston sekä itsenäisesti toimivien olioiden ansiosta sillä voidaan simuloida ja mallintaa monimutkaisia ilmiöitä, kuten liike-elämää ja koulutuksellisia vuorovaikutuksia [Gol95]. Esimerkiksi sairaalan toimintaa voidaan simuloida siten, että kukin potilas, lääkäri ja sairaanhoitaja ovat itsenäisiä olioita, samoin kuin myös yksittäiset sängyt, huoneet, osastot ja leikkaussalit. Oliot pyytävät ja saavat ”palveluita” lähettämällä viestejä toisilleen ja niiden ”tarpeet” ehkä täyttyvät niiden saadessa vastauksen [Sma76].

Simon Lewis kuvaa ohjelmointikielen oppimisprosessia oppimiskäyrällä (Kuva 1) [Lew95]. Jokaisen uuden kielen oppiminen vaatii käyttäjältä aluksi peruskäsitteistön ja syntaksin omaksumista. Epämukavuuden tunne kasvaa niin kauan kuin kielen ominaisuuksia on omaksuttava ymmärtämättä niiden suhdetta toisiinsa tai kokonaisuutta. Kun tiedon taso on riittävä kokonaisuuden ymmärtämiseksi, antavat uudet opitut asiat yhä paremman kuvan koko kielestä ja siten vähentävät oppimiseen liittyvää epämukavuuden tunnetta.



Kuva 1. Smalltalkin oppimiskäyrässä on korkea huippu [Lew95].

Lewis pitää Smalltalkin oppimiskäyrää haastavampana kuin monien muiden kielten. Ensimmäisenä kielenä Smalltalkia opettelevan on aluksi perehdyttävä oliokäsitteeseen sekä selvitettävä metodien, luokkien ja periytyksen merkitys. Vasta tämän jälkeen voi alkaa itse Smalltalkin opettelu. Aikaisemmin ohjelmoinut, jolle oliot saattavat olla ennalta tuttuja, kokee aluksi kulttuurishokin. Kaikki alkeellisimmatkin toiminnot, jotka muissa kielissä kuuluvat syntaksiin, toteutetaan Smalltalkissa täydennettävien ja muokattavan luokkakirjaston olioilla – kieli on olioperustaisuuden läpikäynnin alimmalle tasolle.

## 5 Yhteenveto

Alan C. Kay uskoi 1960-luvulta lähtien tietokoneiden olevan 1990-lukuun mennessä yleisiä kotien, koulujen ja toimistojen laitteita, mikäli kaksi edellytystä täyttyy. Ensimmäinen oli teknisen kehityksen myötä tulevat pienemmät komponentit ja siten pienemmät tietokoneet, toinen oli käyttöliittymien yksinkertaistuminen. Kay ryhtyi 70-luvulla kehittämään tulevaisuuden tietokonetta ja käyttöliittymää Xerox PARCissa apunaan tutkimusryhmä. Smalltalkista oli tarkoitus tulla helppokäyttöinen ja kaikkeen sopiva ohjelmointikieli kannettavaan tietokoneeseen, Dynabookiin. Työn pohjana olivat vuosia hänen mielessään jalostuneet ideat.

Tuloksena oli graafinen olioperustainen järjestelmä, jota ohjataan hiiren ja näytöllä olevien kuvakkeiden avulla. Käyttöliittymästä huolimatta kieli osoittautui niin lapsille kuin aikuisillekin vaikeasti omaksuttavaksi. Xerox ei myöskään uskonut Dynabookin mahdollisuuksiin menestyä markkinoilla. Kay ei itse saanut toteuttaa unelmaansa, mutta hänen ideansa, kuten käyttöliittymän graafisuus, ikkunat ja olio-ohjelmointi jäivät elämään ja kukoistavat yhä käyttämässämme järjestelmissä.

## Lähteet

- Gol95 Adele Goldberg, Why Smalltalk. *Communications of the ACM* 38, 10 (1995), 105-107.
- HPW88 A History of Personal Workstations, toim. Adele Goldberg. ACM Press (1988), 249-263.
- Kay93 Alan C. Kay, The early history of Smalltalk teoksessa History of Programming Languages II, toim. T. Bergin & R. Gibson. ACM Press, Addison-Wesley, USA (1996), 511-598. [Myös ACM Conf. Proc. SIGPLAN Notices 28, 3 (1993), 69-95.]
- Lew95 Simon Lewis, The Art and Science of Smalltalk. Prentice-Hall, 1995.
- Moo65 Gordon Moore, Cramming More Components Onto Integrated Circuits, lehdessä Electronics 38, 8 (19.4.1965). [Myös <ftp://download.intel.com/research/silicon/moorespaper.pdf>, 30.1.2007]
- NyD66 Kristen Nygaard ja Ole-Johan Dahl, Simula – an ALGOL-based simulation language. *Communications of the ACM* 9, 9 (1966), 671-678.
- Sma76 Smalltalk-72 Instruction Manual, toim. Adele Goldberg ja Alan Kay. Xerox Corporation (1976). [Myös [http://www.bitsavers.org/pdf/xerox/alto/Smalltalk72\\_Manual.pdf](http://www.bitsavers.org/pdf/xerox/alto/Smalltalk72_Manual.pdf), 30.1.2007]
- Sol78 Cynthia J. Solomon, Teaching Young Children to Program in a LOGO Turtle Computer Culture. *ACM SIGCUE Outlook* 12, 3 (1978), 20-29.