

T-121.5300 Käyttöliittymäsuunnittelu

Tehtävä 1: Komentorivipohjainen käyttöliittymä

Mikko Vestola
Opiskelijanumero: xxxxx
Sähköposti: xxx
15.11.2007

Sisällysluettelo

1. Johdanto.....	1
2. Suunnittelun lähtötilanne.....	1
3. Toteutettu komentokieli.....	2
4. Käyttöliittymän rakenne ja toiminta.....	3
5. Käyttöliittymän ja käyttäjän välinen vuorovaikutus.....	5
6. Yhteenveto.....	6
7. Palaute työstä.....	6
8. Ohjelmakoodi.....	6

1. Johdanto

Harjoitustyön tarkoituksena oli suunnitella ja toteuttaa yksinkertainen komentokieli DMC-ohjelmistoon. Käytännössä tämä tehtiin toteuttamalla tulkki DMC-ohjelmiston ja käyttäjän käyttämän DMC Browser -dialogi väliin. Tulkki siis ei sinällään tuo ohjelmistoon mitään uutta toiminnallisuutta, vaan yksinkertaistaa DMC:n kieltä käyttäjälle helpommin ymmärrettäväksi ja muuttaa tulosteet käyttäjäystävällisempään muotoon.

Komentokielen avulla käyttäjä voi selata DMC-ohjelmistossa olevia kohdelaitteita ja muuttaa niiden tilaa. DMC-ohjelmistoa käytetään laitteiden etähallintaan. Esimerkiksi kodin TV:n voisi laittaa päälle DMC-ohjelmiston avulla koskematta itse laitteeseen. Tässä harjoitustyössä käyttäjä käyttää DMC:tä valmiin DMC Browser -dialogin avulla. Kyseisen dialogin käyttö ei kuitenkaan ole kovin käyttäjäystävällistä. Toteutettu tulkki asettuikin DMC:n ja dialogin väliin, jolloin käyttäjä voi antaa DMC Browser-dialogille yksinkertaisempia komentoja, jotka tulkki kääntää DMC:n ymmärtämään muotoon.

2. Suunnittelun lähtötilanne

Tässä esimerkissä käyttäjämme on Juho. Juho on jäänyt ylitöihin kiireisen työpäivän vuoksi ja naputtaa yhä tietokoneellaan työtehtäviään. Kun työt alkavat olla päätöksessään, Juho miettii, että haluaisi rankan työpäivän päätteeksi päästä heti rentoutumaan lämpimään saunaan. Mutta saunan lämpiämistä joutuu odottamaan kotiin päästyään puolisen tuntia, mikä ei Juhoa miellytä. Puoli tuntia on kuitenkin juuri se aika, joka Juholta kestää matkata työpaikalta kotiin. Niinpä Juho päättääkin laittaa saunan päälle juuri hankkimansa etäjärjestelmän kautta, jonka hallintaohjelman hän on juuri pari päivää sitten asentanut työkoneeseensa. Juhoa on myös koko päivän kalvanut ajatus, että ei kai hän vain jättänyt valoja päälle eteiseen, kun aamulla lähti töihin. Samalla kun Juho laittaa saunan päälle, haluaisi hän tarkastaa, ovatko eteisen valot päällä. Juho käyttää työkoneettaan ja laittaa sen kautta saunan päälle sekä tarkistaa eteisen valot ennen kun lähtee matkaamaan kohti kotia, jossa pitäisi hänen saavuttuaan odottaa lämmin sauna.

3. Toteutettu komentokieli

Komentokielen toteuttava tulkki on tehty Java-ohjelmointikielellä ja kehitysympäristönä oli Eclipse. Komentokieli koostuu seuraavista komennoista:

- *cd <alue>* (Vaihtaa huonetta tai laitetta. Esim. *cd Bathroom* tai *cd /Living Room/TV*. Mikäli kauttaviivaa ei ole polun alussa, on polku suhteellinen. Kauttaviiva alussa, polku viittaa aina hakemistorakenteen alkuun.)
- *cd..* (Vaihtaa polussa yhden tason alaspäin. Eli jos ollaan esim. huoneessa Hall, voidaan kirjoittaa *cd..* ja päästään takaisin lähtötasoon, jossa on kaikki kodin huoneet.)
- *dir* (Listaa huoneet, huoneen laitteet tai laitteen muuttujat, riippuen missä hakemistossa käyttäjä on komentoa suorittaessa.)
- *ls* (Sama kuin *dir*)
- *help* (Näyttää käyttäjälle tulosteen, josta saa apua komentoihin)
- *set <asetus>=<arvo>* (asettaa laitteen asetuksen annettuun arvoon, esim. *set power=ON*)

Kielen komennot pyrin tekemään Unixeista ja Windowseista löytyvien komentojen kaltaisiksi.

Hyvin luultavaa on, että tämän kaltainen etähallintajärjestelmä on ainakin alkuvaiheessa käytössä tietotekniikkaan perehtyneillä henkilöillä, joille Unixin ja Windowsin komennot ovat tuttuja, joten on hyvä pysyä samantyyolisissä komennoissa, jotka käyttäjä luultavasti tietää ennalta. Kielen suunnittelussa valitsin käyttäjiksi englanninkieliset käyttäjät. Mitään varsinaisia kirjallisia käyttöliittymäohjeistoja en käyttänyt, vaan lähinnä niitä, mitä luennoilla kerrottiin (mm. pyri olemaan yhtenäinen komennoissa) ja mitä itse olen eri käyttötilanteissa kohdannut.

Kun käyttäjä haluaa vaihtaa huonetta, kirjoittaa hän komennon *cd <huone>*, jolloin käyttäjä siirtyy ko. huoneeseen ja eteen tulostuu kaikki huoneen laitteet (tai alahakemistot, joita näytti myös olevan). Huoneen laitteisiin pääsee myös käsiksi samalla komennolla. Komento *cd* siis liikkuu hakemistossa eteen tai taaksepäin (*cd..*), eli käyttäjä ikään kuin liikkuu sisään huoneeseen ja edelleen laitteen äärelle, joka on mielestäni hyvin loogista.

Mielestäni oli järkevää tehdä huoneen vaihto siten, että se listaa aina automaattisesti sen sisältämät laitteet. Automaattilistaus on hyvin muistin tukena, jos käyttäjä ei muista, minkä nimisiä laitteita huoneessa olikaan. Kielen olisi voinut tehdä myös sellaiseksi, että laitteita ei listata automaattisesti (vain käyttäjän hakemisto vaihtuisi) ja käyttäjän olisi pitänyt itse kirjoittaa *dir* tai *ls*, jotta saa laitteet näkyviin. Pienillä laitemäärillä automaattilistaus on kätevä, mutta jos laitteita olisi enemmän, ei se välttämättä olisi järkevä.

Huoneiden listauksissa käytin plus-merkkiä huoneen nimen edessä ja laitteiden listauksissa miinusmerkkiä, jotta huoneet ja laitteet erottuisivat. Muutin myös laitteiden virta-asetuksissa olleet totuusarvot käyttäjälle mielestäni parempaan muotoon: ON / OFF (eikä true / false, joka on ehkä hieman liian ohjelmointikeskeistä). Kommentojen pitäisi myös olla melko virhesietoisia (esim. ylimääräiset välilyönnit eivät haittaa). Kauttaviivoilla polkujen lopussa ei ole väliä, koska mielestäni on vain käyttäjälle hämäävää, että komennot palauttaisivat eri asioita, riippuen siitä, onko polun lopussa kauttaviivaa. Täysin virheellisistä komennoista informoidaan käyttäjää.

4. Käyttöliittymän rakenne ja toiminta

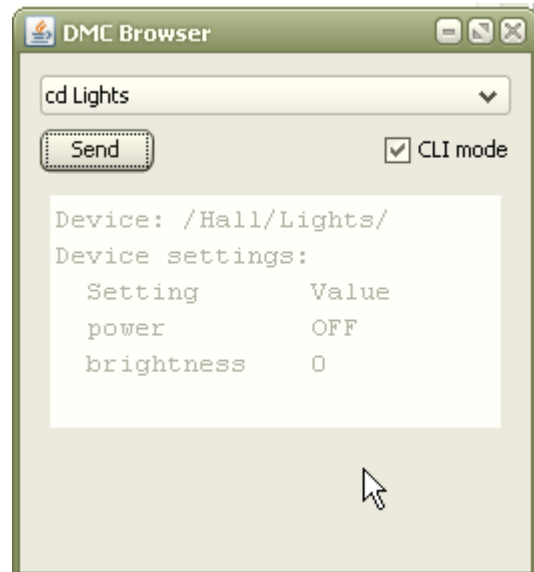
Alla on esitetty käyttöliittymän rakenne ja toiminta esimerkin avulla. Esimerkissä käyttäjä käynnistää ohjelman ja haluaa asettaa eteisen valot päälle. Käyttäjä toimii seuraavan kuvasarjan mukaisesti:



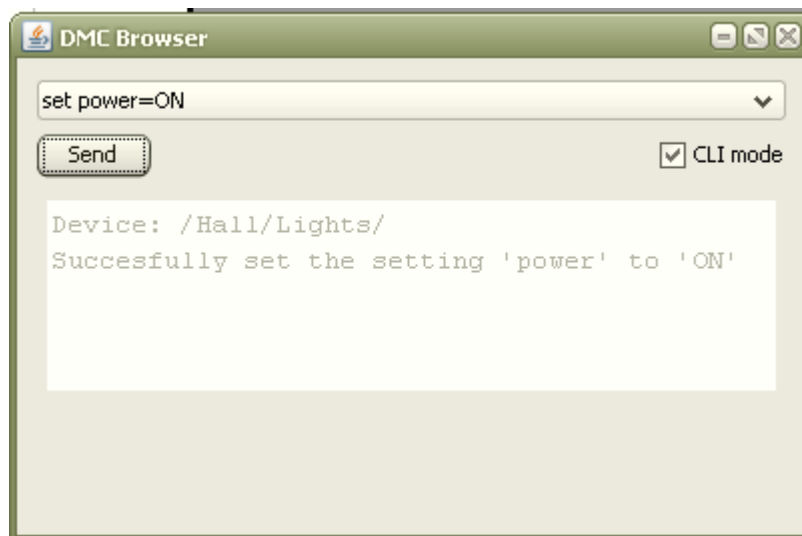
*Kuva 1. Käyttäjä on avannut ohjelman ja kirjoittanut komennon **dir**, **ls** tai **cd /**. Nämä kaikki listaavat järjestelmässä olevat huoneet käyttäjän ikkunaan. Tulosteen 1. rivillä näkyy aina senhetkinen polku (eli huone tai laite). Seuraavalla rivillä näkyvät listaukset huoneista, laitteista tai asetuksista, riippuen, missä hakemistossa ollaan.*



Kuva 2. Käyttäjä on siirtynyt eteiseen (Hall) komennolla **cd Hall** ja näkee listauksen huoneessa olevista laitteista.



Kuva 3. Käyttäjä vaihtaa eteisessä (Hall) olevaan valoihin (laite Lights) komennolla **cd Lights** ja näkee valojen senhetkiset asetukset.



Kuva 4. Käyttäjä asettaa eteisen (huone Hall) valot (laite Lights) päälle antamalla komennon **set power=ON**. Ohjelma palauttaa käyttäjälle viestin, että toiminto onnistui.

5. Käyttöliittymän ja käyttäjän välinen vuorovaikutus

(Huom. Tässä oletetaan, että käytössä on ”oikea” komentorivi, eli komento suoritetaan painamalla enter-näppäintä, toisin kuin DMC Browser-ohjelmistossa on tällä hetkellä)

Juho haluaa laittaa saunansa lämpiämään työpaikaltaan käsin, jotta pääsee kotiin saavuttuaan heti lämpimään saunaan. Hän istuu käynnissä olevan työkoneensa ääreen ja avaa DMC Browser-ohjelman kaksoisklikkaamalla työpöydällä olevaa etähallintajärjestelmän kuvaketta.

Ohjelma aukeaa ja eteen tulee komentorivi, johon Juhon pitäisi kirjoittaa komentoja. Juho ei kunnolla muista, että mikä komento teki mitäkin, mutta hän muistaa, että komennolla *help* hän saa tietoa käytettävistä komennoista. Juho kirjoittaa *help* ja painaa enteriä (kuten jokaisen muunkin komennon suorittamisen kohdalla), jolloin ikkunaan tulostuvat ohjeet käytettävissä olevista komennoista. Nyt Juho muistaa, mikä komento teki mitäkin. Juho kirjoittaa *dir*, jolloin kaikki hänen kotinsa huoneet listautuvat ikkunaan. Listassa näkyy huone *Bathroom* ja Juho tietää, että sauna, jonka hän haluaa päälle, on kyseisessä huoneessa. Hän haluaa siirtyä ko. huoneeseen ja kirjoittaakin komennon *cd Bathroom*, jolloin ikkunaan tulostuu pesuhuoneessa olevat laitteet. Listalla on laite nimeltä *Sauna*, jolloin Juho kirjoittaa *cd Sauna*.

Juho näkee nyt ikkunassa saunan asetukset. Valittavia asetuksia ovat **power** (kiukaan virta) ja **maxTemp** (lämpötila, johon termostaatti lämmityksen lopettaa). Asetuksissa näkyvät arvot **power: OFF** ja **maxTemp: 60**. Lämpötila on se mihin Juho on totunut, joten siihen hän ei koske. Sen sijaan hän haluaa laittaa kiukaan virran päälle, joten hän kirjoittaa komennon *set power=ON*. Ikkunaan tulostuu ilmoitus, joka kertoo, että asetus muutettiin onnistuneesti. Juho tietää nyt, että kiuas on päällä.

Juho muistaa, että hänen piti myös tarkistaa, ovatko eteisen valot jääneet päälle. Hän muistaa tehneensä kyseisen operaation jo kerran aikaisemminkin, ja muistaakin, mistä asetus löytyy. Juho kirjoittaa komennon *cd /Hall/Lights*, jolloin ikkunaan tulostuu eteisen valojen tila. Juho huomaa, että ikkunassa valojen asetuksen **power** arvo on **ON**. Hän siis jätti kuin jättikin valot päälle. Pikapikaa Juho kirjoittaa komennon *set power=OFF*, jolloin ikkunaan tulostuu tieto, että eteisen valot on kytketty pois päältä. Juho sulkee lopulta DMC Browser-ohjelman oikean yläkulman ruksi-painikkeesta, sammuttaa koneensa ja lähtee kohti kotona odottavaa lämpenevää sauna.

6. Yhteenveto

Mielestäni toteutettu komentokieli onnistui melko hyvin. Siinä on vielä hiomista (esim. help-komennon palauttama ohjeteksti ja muutamia virhetilanteita), mutta on ainakin käytettävyydeltään paljon parempi kuin DMC Browserin nykyinen käyttöliittymä, jossa käyttäjän täytyy muistaa hyvinkin kryptisiä komentoja. Omasta mielestäni on hyvin loogista käyttää kielessä rakennetta, että käyttäjä siirtyy cd-komennolla eri huoneiden välillä ja edelleen ikään kuin laitteiden ääreen ja näkee sitten laitteen asetukset. Tämä vastaa ehkä parhaiten fyysistä tilannetta, joka on käyttäjälle tuttua.

7. Palaute työstä

Työ oli mielestäni ihan mielenkiintoinen. Aluksi oli hieman vaikea saada käsitystä, mitä käytännössä piti toteuttaa. Aluksi ymmärsin, että olisi pitänyt tehdä esimerkiksi Windowsin komentorivillä toimiva käyttöliittymä, mutta luennoilla selvisi lopulta, että kyseessä olikin vain tulkki DMC Browseriin. Loppujen lopuksi tehtävä oli mielestäni melko helppo. Tosin aikaa tähän kului kyllä enemmän kuin se varattu 6h (ehkä lähemmältä 10h). Itselläni ei tuhraantunut aikaa kehitysympäristön asentamiseen, koska Eclipsen käyttö ja Java olivat hyvin tuttuja. DMC:kin toimi hienosti heti ensi yrittämällä eikä ongelmia juuri ollut. Niille, joille kehitysympäristöt ja ohjelmointikielet eivät ole tuttuja, kuluu kyllä varmasti yli tuo 6h. Mutta omasta mielestäni tehtävä oli ensimmäiseksi tehtäväksi sopivan tasoinen. Itse DMC:kin vaikuttaa hyvin mielenkiintoiselta ja täytyykin varmaan tutustua siihen, kun lähdekoodit julkaistaan.

8. Ohjelmakoodi

Ohjelmakoodi on liitteenä erillisessä tiedostossa [CmdLineHandler.java](#). Koodista olisi tullut niin sotkuista, jos olisin sen raportin loppuun liittänyt, että näin paremmaksi liittää sen erilliseen tiedostoon.