

Merkkauskielten käyttö puhesynteesissä

Minna Kamppuri
61005W

28.05.2001

Joensuun yliopisto
Tietojenkäsittelytiede
Kandidaatintutkielma

Tiivistelmä

Puhesynteesin tavoitteena on saada aikaan mahdollisimman ymmärrettävää ja miellyttävän kuuloista puhetta keinotekoisesti tietokoneen avulla. Tekstistä puheeksi-järjestelmissä yksi puhesynteesin ongelmista on se, miten kone löytäisi kirjoitetusta tekstistä kaiken puheen tuottamisen kannalta tärkeän informaation. Yhdeksi ratkaisuksi tähän ongelmaan on ehdotettu merkkauskieliä. SABLE on yksi viimeisimmistä tätä tehtävää varten kehitetyistä merkkauskielistä. Tässä tutkielmassa esittelemme lyhyesti luonnollisia kieliä ja merkkauskieliä sekä perehdymme SABLE:n mahdollisuuksiin puhesynteesin merkkauskielenä kirjallisuuden ja pienimuotoisen käytännön kokeilun avulla.

ACM-luokat (ACM Computing Classification System, 1998 version): I.2.7, I.7.2

Avainsanat: merkkauskielet, puhesynteesi, SABLE

Sisältö

| | |
|---|-----------|
| 1 Johdanto | 1 |
| 2 Puhesynteesi | 2 |
| 2.1 Luonnolliset kielet | 2 |
| 2.2 Yleistä puhesynteesistä | 6 |
| 2.3 Tekstistä puheeksi -järjestelmät | 7 |
| 3 Merkkaukielet | 11 |
| 3.1 Merkkaukielet puhesynteesissä | 11 |
| 3.2 Merkkau ja rakenteiset dokumentit | 12 |
| 3.3 XML | 13 |
| 4 SABLE | 19 |
| 4.1 Puhesynteesiä varten kehitellyt merkkaukielet | 19 |
| 4.2 SABLE | 20 |
| 4.3 SABLE:n käyttäminen synteesijärjestelmissä | 26 |
| 4.4 SABLE:n arviointi | 28 |
| 5 Yhteenveto | 31 |
| Viitteet | 33 |
| Liite 1: SABLE:n dokumenttityypinmäärittely | 36 |
| Liite 2: Syntetisoidut tekstit ilman tunnisteita | 41 |
| Liite 3: Syntetisoidut tekstit tunnisteilla merkattuna | 43 |
| Liite 4: Levyke | 46 |

1 Johdanto

Puhesynteesiä on tehty jo kymmenien vuosien ajan, mutta vasta viime aikoina syntetisoidusta puheesta on tullut suosittu niin tutkimuksen kuin kaupallisten sovellustenkin puolella. Syynä puhesynteesin äkilliseen suosioon on varmasti matkapuhelimien määrän räjähdysmäinen kasvu, mutta myös se, että puhesyntetisaattoreiden puhe on aivan näihin päiviin asti ollut robottimaista ja luonnotonta. Tämä johtuu siitä, että ihmisen puhe on hyvin monimutkainen prosessi, jonka jäljittely vaatii runsaasti tietoa, tutkimusta ja tietokonekapasiteettia.

Nykyiset koneet ovat jo riittävän tehokkaita suoriutuakseen puheen tuottamiseen vaadittavasta työmäärästä. Ongelmana on kuitenkin se, etteivät koneet pysty ”ymmärtämään” lukemaansa tekstiä. Monet ihmisen puheeseen vaikuttavat seikat, jotka liittyvät kielen merkityksiin ja tilanteisiin, ovat tekstiä analysoiville koneille lähes mahdottomia löytää.

Merkkauskieliä on käytetty dokumenttien rakenteiden kuvaamiseen 1980-luvulta lähtien, jolloin SGML-standardi otettiin käyttöön (Light, 1997). Myöhemmin SGML:n ja siihen perustuvan XML:n avulla on kehitetty merkkkauskieliä, joita voidaan käyttää antamaan puhesynteesijärjestelmälle lisätietoa syntetisoidavasta tekstistä ja parantamaan näin puhesynteesin laatua.

SABLE (SABLE: A Synthesis Markup Language, 2000) on yksi tällaisista merkkkauskielistä. Se on useiden aikaisemmin samaan tarkoitukseen kehiteltyjen merkkkauskielten yhdistelmä, josta pyritään luomaan yleistä standardia puhesynteesin alalle. SABLE:a kehitetään yhä, sillä vielä ei tarkkaan tiedetä, mitä kaikkia asioita teksteistä olisi pystyttävä kuvaamaan riittävän hyvän puhesynteesin aikaansaamiseksi.

Tämän tutkielman tarkoituksena on antaa lukijalle käsitys puhesynteesiprosessista, merkkkauskielistä ja siitä, missä määrin merkkkauskielten käyttö voi ratkaista luonnollisen kielen mallintamiseen liittyviä ongelmia puhesynteesissä. Luku 2 alkaa pienellä katsauksella luonnolliseen kieleen ja sen rakenteeseen, jonka jälkeen

siirrymme tarkastelemaan puhesynteesiä ja yhtä puhesynteesijärjestelmien päälajeista eli tekstistä puheeksi-järjestelmiä. Luvussa 3 käymme läpi merkkaukielten periaatteita ja tutustumme XML-standardiin, johon SABLE:kin pohjautuu.

Luvussa 4 perehdymme tarkemmin esimerkkien valossa SABLE:en ja sen käyttömahdollisuuksiin. Pienenä käytännön esimerkkinä otamme kaksi tekstidokumenttia ja tarkastelemme, kuinka paljon SABLE:n avulla on mahdollista vaikuttaa siihen, miten tekstistä puheeksi-järjestelmä tekstin tulkitsee. Tutkielman lopuksi esitämme lyhyen yhteenvedon.

2 Puhesynteesi

Puhesynteesiprosessin ymmärtäminen vaatii jonkun verran tietämystä luonnollisen kielen ominaisuuksista, minkä vuoksi tutustummekin aluksi luonnollisen kielen rakenteeseen ja muutamiin kielitieteen termeihin. Kuvien esimerkit ovat kirjoittajan.

2.1 Luonnolliset kielet

Luonnolliset kielet ovat monitasoisia, erittäin kompleksisia järjestelmiä (Karlsson, 1998). Ne koostuvat erilaisista osajärjestelmistä, jotka käymme seuraavaksi lyhyesti läpi.

Fonologia on kielen äännerakenteen tutkimista (Karlsson, 1998). Ihmisen puheessa on lukematon määrä äännteitä eli *fooneja*, joissa esiintyy runsaasti yksilöllistä vaihtelua. *Foneemit* ovat äännteiden abstraktioita, ja äännteet voivat olla joko saman foneemin tai eri foneemien edustajia eli *allofoneja*. Foneemi siis reaalistuu puheessa jonkin verran vaihtelevan asuisena.

Kuvassa 1 esitetty IPA:n (International Phonetic Association) aakkoston mukainen karkea foneettinen transkriptio on lähinnä foneemitasoa. Siitä ei käy ilmi esi-

merkiksi se, että eri äänneympäristöistä johtuen [k] sanassa ”kissa” on todellisuudessa hyvinkin erilainen kuin [k] sanassa ”kummallisesti” (Karlsson, 1998): niitä voidaan kuitenkin pitää saman foneemin /k/ allofoneina.

"Tuo kissa käyttäytyy kummallisesti."

[t u o k i s : a k a e y t : a e y t y : k u m : a l : i s e s t i]

Kuva 1: Lause ”Tuo kissa käyttäytyy kummallisesti” foneettisena transkriptiona.

Puheessa esiintyvät vokaalit ja konsonantit ovat puheen *segmentaalisia ominaisuuksia*, kun taas *suprasegmentaalisia* eli *prosodisia ominaisuuksia* ovat pituus, painotus ja intonaatio (Karlsson, 1998). *Pituutta* voidaan käyttää sanojen merkityksen erottamiseen, kuten esimerkiksi suomen sanoissa ”kisa” - ”kissa”. *Painolla* tarkoitetaan sitä, että jokin puheessa esiintyvä tavu tai sana äännetään siten, että se erottuu voimakkaammaksi kuin ympäristön muut tavut tai sanat (Wiik, 1981). Painon tunnusmerkkejä voivat olla suurempi äänen voimakkuus, korkeampi sävelkorkeus (perusvärähtely), pitempi kesto sekä selkeämpi ääntäminen. Näitä ominaisuuksia käytetään samankin kielen sisällä eri suhteissa tilanteesta riippuen.

Sanapaino voi osoittaa sananrajoja: näin on esimerkiksi suomessa, jossa sanan pääpaino on aina ensimmäisellä tavulla (Karlsson, 1998). *Lausepainon* suhteen useimmat *puhunnokset* eli sellaiset yhtäjaksoiset puheen pätkät, joilla on fyysisesti todettavat rajat, saavat *neutraalipainotuksen*. Hyvin usein kielissä esiintyy *remaattista painotusta*, jolloin voimakkain paino kohdistuu siihen sanaan, joka ilmaisee keskeisintä uutta informaatiota lauseessa. *Kontrastiivisella painotuksella* voidaan puolestaan tähdentää yhtä vaihtoehtoa.

Kuvan 2 ensimmäisessä lauseessa käytetään remaattista painotusta, jolloin paino tulee lauseen lopussa sijaitsevaan kissan käytöksen laatua ilmaisevaan sanaan. Toisessa lauseessa on kontrastiivinen painotus, jonka avulla halutaan kertoa, että puhutaan nimenomaan tietystä kissasta eikä esimerkiksi koirasta.

"Tuo kissa käyttäytyy KUMmallisesti."

"Tuo KISSa käyttäytyy kummallisesti."

Kuva 2: Remaattinen ja kontrastiivinen painotus.

Intonaatiolla tarkoitetaan puheessa esiintyvää sävelkulkua (Karlsson, 1998). Lausetasolla intonaatio voi osoittaa sanojen tai lauseiden kuulumista yhteen. Lisäksi *intonaatiokontuurilla* eli yhtäjaksoisella sävelkulkukokonaisuudella on ekspressiivisiä ja kieliopillisia tehtäviä, kuten kysymyslauseen erottaminen väitelauseesta.

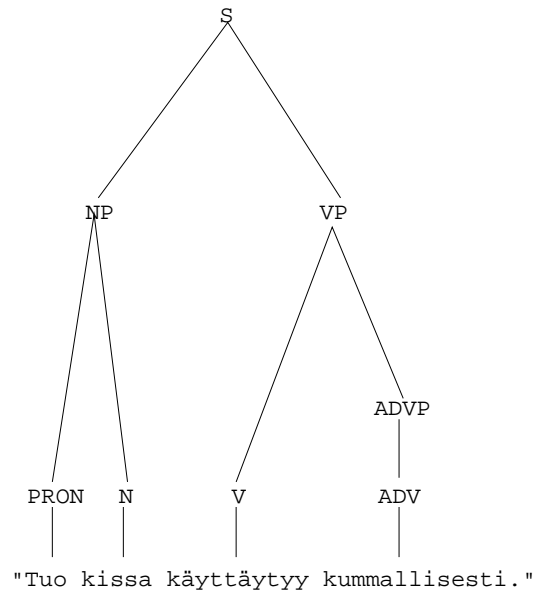
Morfologia eli muoto-oppi tarkoittaa sanojen sisäisen rakenteen osajärjestelmää (Karlsson, 1998). *Morfeemi* on kielen pienin merkityksenkantaja. Aivan kuten foneemi, morfeemikin on eräänlainen abstraktio, joka käytännössä toteutuu eri *morfeina*. Tällöin esimerkiksi sanoissa "talossa" ja "kylässä" sanojen morfit "-ssa" ja "-ssä" ovat saman morfeemin *allomorfeja*. Kuvan 3 lauseessa "-y" osoittaa kolmatta persoonaa, "-llise" on adjektiivipääte ja "-sti" tekee adjektiivista adverbien.

"Tuo kissa käyttäyty+y kumma+llise+sti."

Kuva 3: Esimerkkilauseen sisältämät morfeemit.

Leksikko eli sanavarasto on vakiintuneiden sanojen osajärjestelmä (Karlsson, 1998). *Lekseemit* ovat yhteen kuuluvien sanamuotoesiintymien luokka, ts. esimerkiksi suomen sanamuodot "varvas", "varpaan" ja "varpaisillaan" edustavat samaa lekseemiä. Sanat voidaan jakaa *sisältösanoihin* ja *funktiosanoihin* (O'Grady, 1997): sisältösanat ovat substantiiveja, verbejä ja adjektiiveja, kun taas funktiosanat ovat kieliopillisia sanoja kuten pronomineja ja prepositioita.

Seuraava kielen taso on *syntaksi* eli lauseoppi (Karlsson, 1998). Se tutkii lauseiden rakentumista sanoista. Lauseilla on hierarkkinen rakenne, jota voidaan kuvata erilaisilla puukuvaimilla, joihin merkitään lausekkeet ja sanaluokat.



Kuva 4: Esimerkkilauseesta ”Tuo kissa käyttäytyy kummallisesti” tehty puukuvain.

Fonologia, morfologia, leksikko ja syntaksi ovat muodollisia osajärjestelmiä, joilla on aineellinen muoto (Karlsson, 1998). *Semantiikka* tutkii sen sijaan kielen merkityksiä, joilla ei ole näkyvää muotoa, vaan jotka hahmottuvat vastaanottajan tulkintaprosessissa. Kielen merkitystä on vaikea määritellä: sen luonteesta ja siitä, miten se esiintyy ihmisen mielessä tiedetään yhä varsin vähän (O’Grady, 1997). Sanojen ja lauseiden välillä on erilaisia semanttisia suhteita kuten *homonymia* ja *polysemia*, jotka viittaavat muotoihin, joilla on useita merkityksiä (esim. ”kuusi” voi olla luku, puulaji tai yksikön toisen persoonan omistusmuoto sanasta ”kuu”).

Semantiikkaan kuuluu läheisesti *pragmatiikka*, joka tutkii kielen käyttöön ja ym-

märtämiseen liittyviä tekijöitä (O’Grady, 1997). Pragmatiikan tutkimuskohteita ovat mm. puhujan ja kuulijan asenteet ja uskomukset, heidän käsityksensä lauseen kontekstista ja tietämyksensä siitä, kuinka kieltä voidaan käyttää informaation välittämiseen, suostutteluun, harhauttamiseen ja niin edelleen. Puhujan uskomukset heijastuvat koko ajan puheeseen: esimerkiksi kysymys ”Oletko lopettanut kuntoilun?” sisältää implisiittisen oletuksen siitä, että henkilö, jolta tätä kysytään, on jossain vaiheessa harrastanut kuntoilua.

Yhdeksi kielen tutkimuksen alueeksi voidaan laskea myös *diskurssianalyysi*, jonka keskeisiä kohteita ovat eri keskustelu- ja puhetyyppien ominaispiirteet (Karls-son, 1998). Näitä piirteitä tutkitaan keskusteluista tehtyjen transkriptioiden avulla (O’Grady, 1997).

2.2 Yleistä puhesynteesistä

Puhsynteesillä tarkoitetaan puheen tuottamista keinotekoisesti tietokoneen avulla. Puhsynteesi voi olla joko interaktiivista tai ei-interaktiivista riippuen siitä, missä määrin käyttäjä on vuorovaikutuksessa järjestelmän kanssa (Bhaskarao, 1994). Interaktiivisessa puhsynteesissä järjestelmä muuntaa saadun syötteen puheeksi lähes reaaliajassa, kun taas ei-interaktiivista puhsynteesiä käytetään esimerkiksi automaattisissa kuulutusjärjestelmissä, joissa käyttäjä ei juurikaan pääse vaikuttamaan järjestelmän toimintaan.

Puhsynteesijärjestelmät voidaan jakaa kahteen eri ryhmään niiden syötteen muodon perusteella (Pfister & Traber, 1994). Niin sanotuissa *concept-to-speech* -järjestelmissä puhe syntetisoidaan erilaisten semanttisten tai pragmaattisten konseptien pohjalta, jolloin järjestelmä on selvillä syntetisoitavien puhunnosten merkityksestä ja tarkoituksesta. Tällaista puhsynteesiä käytetään silloin, kun järjestelmä luo itse puhunnosten sisällön. Näin tapahtuu esimerkiksi automaattisissa dialogijärjestelmissä, joissa järjestelmä muotoilee vastaukset käyttäjien kyselyihin.

Toisen puhesynteesijärjestelmien ryhmän muodostavat *tekstistä puheeksi* - eli *TP-järjestelmät* (text-to-speech systems, TTS-systems) (Pfister & Traber, 1994). Nämä järjestelmät saavat syötteenään tekstiä, jonka ne muuntavat puheeksi jäljitellen näin ääneen lukemista. Suoriutuakseen tästä tehtävästä TP-järjestelmän täytyy saada selville esimerkiksi luettavan tekstin kielellinen rakenne. TP-järjestelmiä ovat muun muassa näkövammaisille tarkoitettut lukulaitteet sekä automaattiset tietopalvelut, joiden avulla on mahdollista kuunnella puhelimen välityksellä tuoreimmat uutiset tai saapuneet sähköpostit. Tämän tutkielman rajallisuuden vuoksi keskitymme jatkossa tarkastelemaan puhesynteesiä pelkästään TP-järjestelmissä.

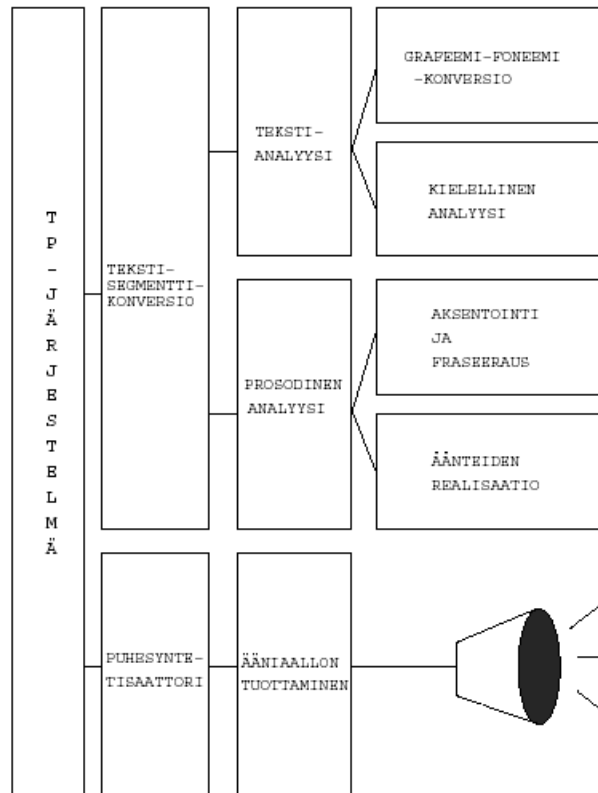
2.3 Tekstistä puheeksi -järjestelmät

Tekstin muuntaminen puheeksi vaatii TP-järjestelmältä syötetekstin analysoimista monella eri tasolla. Eri järjestelmien arkkitehtuurit vaihtelevat (Pfister & Traber, 1994), joten kuvassa 5 esitellään yleisellä tasolla ne vaiheet, joita korkeatasoisen puhesynteesin tuottaminen tekstistä vaatii. Käymme seuraavaksi nämä vaiheet tarkemmin läpi: aloitamme kielen matalimmalta tasolta eli ääniaalloista, ja tarkastelemme, mitä syötteitä kukin synteesin taso vaatii voidakseen suorittaa vaaditut tehtävät.

Ääniaallon tuottaminen

Puheen tuottamiseen on olemassa useita eri metodeja (Pfister & Traber, 1994). *Artikulatoriset mallit* (articulatory models) tuottavat puhesignaaleja jäljittelemällä ihmisen ääntöelinten akustisia ominaisuuksia. *Puhesignaalimallit* (speech signal models) puolestaan yrittävät jäljitellä puheen havaitsemisen kannalta olennaisia puhesignaalin ominaisuuksia kuten formantteja.

Kolmas puheen tuottamisen malli on *konkatenatiivinen malli* (concatenative model) (Pfister & Traber, 1994). Tässä mallissa puhe syntyy yhdistelemällä luonnollisesta puheesta äänitettyjä ja leikattuja segmenttejä. Yhdisteltävät segmentit ovat tavallisimmin yhden tai kahden äänteen mittaisia.



Kuva 5: TP-järjestelmän osat (Pfister & Traber, 1994 sekä Bhaskararao, 1994 pohjalta)

Puheen tuottamisesta huolehtivan komponentin täytyy tietää, mitä äänneitä tuotetaan ja myös se, miten pitkinä, millä sävelkorkeudella ja millä voimakkuudella nämä äänneet tuotetaan (Pfister & Traber, 1994). Toisin sanoen puhesynteesin tässä vaiheessa puhesynteesijärjestelmä tarvitsee sekä tietoa siitä, kuinka syntetisoitava teksti toteutuu foneemeina että prosodisia parametreja.

Prosodinen analyysi

Saadakseen selville, miten syntetisoitava teksti toteutuu foneemeina, järjestelmä tarvitsee abstraktin kuvauksen syntetisoitavista äännejonoista. Tähän abstraktiin kuvaukseen sovelletaan sitten niitä muunnoksia, joita esimerkiksi äänneiden konteksti, käytettävä murre ja puhujan tyyli voivat saada aikaan äännejonoissa (Pfister & Traber, 1994).

Segmentaalisten ominaisuuksien lisäksi puheen tuottamisessa tulee ottaa huomioon prosodiset ominaisuudet. Nämä ominaisuudet voivat olla sekä kieli- että murrekohtaisia, mutta myös joitakin universaaleja, kaikille maailman kielille yhteisiä piirteitä löytyy (Pfister & Traber, 1994). Tällainen piirre on esimerkiksi intonaation eli sävelkorkeutta ilmaisevan F0-käyrän tasainen lasku puhunnoksen aikana.

Kielen sisäiset prosodiset ominaisuudet ovat pitkälti riippuvaisia puhunnosten aksentoinnista ja *fraseerauksesta* eli lauseissa esiintyvien sanojen ryhmittelystä (Pfister & Traber, 1994). TP-järjestelmä kiinnittää sopivan painon jokaiselle puhunnoksen tavulle. Tätä varten järjestelmä tarvitsee tietoa sanapainojen ja lausepainojen sijainnista. Lausepainot määräytyvät suurelta osin tekstin pragmaattisen ja semanttisen sisällön perusteella, joita järjestelmän on hyvin vaikeaa, ellei mahdollonta, löytää tekstistä. Jotain tästä korkeamman tason kielellisestä tiedosta on kuitenkin löydettävissä lauseiden leksikaalisesta ja syntaktisesta rakenteesta. Esimerkiksi funktiosanat ovat yleensä painottomampia kuin sisältösanat.

Lauseiden fraseeraus riippuu pitkälti puhunnoksen syntaktisesta rakenteesta (Pfister & Traber, 1994). Painotus ja fraseeraus toteutetaan TP-järjestelmissä sääntöjen

avulla, jotka puolestaan pohjautuvat useimmiten kielelliseen tietoon.

Tekstianalyysi

Äännejonojen todellinen esiintymisasu saadaan muuntelemalla niiden abstraktia muotoa. Tämä abstrakti muoto on tehtävä syötetekstistä, joka on kirjoitetussa muodossa (Pfister & Traber, 1994). Muunnosta kutsutaan *grafeemi-foneemi-konversioksi*. Nykyisten koneiden muistikapasiteetin jatkuvasti kasvaessa on siirrytty käyttämään suuria sanakirjoja, jonne voidaan tallettaa sanojen ääntämykset. Poikkeukset, joita ei sanakirjasta löydy, on järjestelmän kuitenkin käsiteltävä sääntöjen avulla. Nämä säännöt voivat olla järjestettyjä joukkoja kontekstisia uudelleenkirjoitussääntöjä (Pfister & Traber, 1994).

Aksentoinnissa tarvittava kielellinen tieto koostuu syntaktisesta, leksikaalisesta ja morfologisesta analyysistä (Pfister & Traber, 1994). Syntaktisen analyysin tarkoituksena on löytää jokaisen lauseen ns. *hierakkinen konstituenttirakenne*, joka vaikuttaa lausepainoon ja fraseeraukseen. Lisäksi syntaktinen analyysi tukee myös tekstin *normalisointia* eli oikean ääntämyksen löytämistä homografeille, lyhenneille ja numeroille. Syntaktista analyysia varten tarvitaan jokaisen sanan syntaktinen kategoria sekä tietoa niiden sijasta, luvusta, suvusta jne. eli leksikaalista analyysia. Mikäli käytössä olevassa kielessä on rikas taivutus, ei ole käytännöllistä tallettaa kaikkia sanamuotoja sanakirjaan. Tällöin tarvitaan lisäksi morfologista analyysia, joka ns. *allomorfileksikon* ja oman kieliopin avulla määrittelee, miten sanat jakautuvat morfeemeihin.

Käytännössä kaikki TP-järjestelmät tekevät morfologista ja syntaktista analyysia ainakin jollakin tasolla (Pfister & Traber, 1994). Eroja löytyy sanakirjojen koossa ja analyysin täydellisyydessä: monissa järjestelmissä pyritään löytämään lauseista vain prosodisesti kaikkein tärkeimmät konstituentit.

3 Merkkauskielet

Tässä luvussa pohdimme aluksi, miten merkkauskielen avulla voidaan parantaa puhesynteesin laatua ja tutustumme sitten merkkauskielten yleisiin ominaisuuksiin. Luvun loppuksi esittelemme XML:n sillä tasolla kuin se on SABLE:n ja erityisesti tutkielman liitteenä 1 olevan SABLE:n dokumenttityyppimäärittelyn ymmärtämisen kannalta tarpeellista.

3.1 Merkkauskielet puhesynteesissä

Pystyäkseen todella jäljittelemään ihmistä tekstin lukemisessa, TP-järjestelmän tuottaman puheen tulisi olla mahdollisimman luonnollista ja ymmärrettävää. Puheen luonnollisuus on ratkaisevasti kiinni puheen prosodiasta, jolla on havaittu olevan myös puheen ymmärrettävyyteen jopa segmenttien laatua ratkaisevampi merkitys.

Sopivien prosodisten parametrien löytäminen edellyttää kuitenkin myös sellaista korkeamman tason kielellistä analyysia, jonka tekeminen ei ole nykyisillä TP-järjestelmillä mahdollista (Wouters & al, 1999). Prosodia on riippuvainen useista eri seikoista, kuten lauseen merkityksestä ja puhujan ominaisuuksista ja tunnetiloista. Tällaisista asioista on kirjoitetussa tekstissä hyvin vähän sellaisia vihjeitä, jotka olisivat löydettävissä automaattisen analyysin avulla (Lemmetty, 1999).

Merkkauskielen avulla voimme antaa järjestelmälle tietoa sellaisista syötetekstin ominaisuuksista, joilla on merkitystä synteesiä tehtäessä, mutta joita järjestelmän on vaikeaa löytää tekstistä. Näin voimme disambiguoida moniselitteisiä sanoja tai lauseita tai osoittaa esimerkiksi korostuksen avulla tekstin sisältämiä pieniä merkitysvivahteita. Järjestelmälle voidaan antaa tietoa poikkeuksellisten sanojen tai lyhenteiden oikeasta ääntämyksestä, ja tarvittaessa järjestelmää voidaan ohjata prosodiaparametrien tasolla määrittelemällä suoraan käytettävää puhenopeutta, sävelkorkeutta, äänen voimakkuutta jne.

3.2 Merkkkaus ja rakenteiset dokumentit

Termillä *merkkkaus* (markup) tarkoitetaan mitä tahansa dokumentin tekstiin lisättyä informaatiota. Sitä käytetään mm. tekstinkäsittelyohjelmissa kontrolloimaan tekstin ulkoasua kuten eri kirjasimien valintaa tai lihavoitinta (Light, 1997). Alla on esimerkki RTF-muotoisessa tekstissä käytettävästä merkkkauksesta:

```
{\rtf1\ansi\ansicpg1252\deff0deflang1035{\fonttbl
{\f0\fnil\fcharset 0 Times New Roman;} }
\uc1\pard\u\none\f0\fs20EsimerkkiRTF-muodosta.\par}
```

Toista merkkkauksen lajia edustavat *rakenteiset dokumentit* (structured documents): niissä keskitytään erottamaan dokumentin rakenne ja sen sisältämä varsinainen informaatio toisistaan (Immosen (1999) mukaan Salminen, 1992). Rakenteiset dokumentit koostuvat *elementeistä*, joita merkitään *tunnisteilla*. Seuraavassa on pieni esimerkki tällaisesta dokumentista:

```
<viesti><lahettaja>Aino Ailas</lahettaja>
      <vastaanottaja>Kalle Kailas</vastaanottaja>
      <sisalto>Palaveri siirtyy huomiseen!</sisalto>
</viesti>
```

SGML (Standard Generalized Markup Language) on kansainvälinen dokumenttirakenteiden määrittelyyn tarkoitettu standardi. Se sisältää *metakielen*, jonka avulla voidaan muodostaa dokumenttien loogisen rakenteen kuvaava kielioppi eli kuvata merkkkauskieli (Immonen, 1999). SGML:n avulla voidaan esittää dokumenttien rakenne: sen hyödyntämiseen tarvitaan lisäksi ohjelmia, jotka ymmärtävät SGML:ää ja käsittelevät dokumentteja käyttäjän haluamalla tavalla (Light, 1997). SGML:ää on käytetty laajalti erilaisten dokumenttien rakenteiden kuvaukseen esimerkiksi sotilasilmailun, kustannusteollisuuden ja tietoliikenneteollisuuden aloilla (Immonen, 1999).

SGML-dokumentti koostuu tavallisesti esittelyosasta, dokumenttityypimäärittelystä (document type definition, DTD) ja dokumentin instanssista. *Esittelyosa* kertoo dokumenttityypimäärittelyn konkreettisen syntaksin, *dokumenttityypimäärittely* kuvaa ne rakenteelliset säännöt, joita dokumentin merkkauksen tulee noudattaa, ja *dokumentin instanssi* on dokumentin SGML-merkattu sisältö (Immonen, 1999).

3.3 XML

SGML on alustariippumaton, joustava ja paljon käytetty standardi, mutta se on myös laaja ja monimutkainen. Niinpä SGML:stä on kehitetty myös yksinkertaisempi versio, *XML* (eXtensible Markup Language). XML sisältää vain pienen osan kaikista SGML:n ominaisuuksista, mutta suurimman osan sen hyödyistä (Light, 1997).

XML:n ominaisuudet

Kuten SGML, myös XML mahdollistaa omien tunnistejoukkojen määrittämisen, joten dokumentteja voidaan räätälöidä sen mukaan, mihin niitä tarvitaan ja millaista informaatiota niistä täytyy käsitellä. XML-dokumentit voivat olla joko valideja tai hyvin muodostuneita dokumentteja. *Validi dokumentti* noudattaa kaikkia dokumenttityypimäärittelyssään olevia sääntöjä. *Hyvin muodostunut dokumentti* ei välttämättä ole dokumenttityypimäärittelynsä mukainen, mutta noudattaa kuitenkin kahta XML:n perussääntöä (Light, 1997):

- kaikilla elementeillä on sekä alku- että lopputunnisteet ja
- kaikki elementit ovat kokonaan sisäkkäin tai peräkkäin eivätkä limity keskenään.

Kaikilla XML-dokumenteilla on siten hyvin selkeä sisäinen rakenne, mikä puolestaan auttaa informaation mallintamisessa ja hakemisessa (Light, 1997).

XML:ssä on myös hyperlinkki- ja tyylimekanismit. XML:n linkit voivat osoittaa erilaisiin lähteisiin ja olla monisuuntaisia. Tyylin määrittämiseen käytetään XML:ssä tyylisivuja ja XSL:ää (eXtensible Stylesheet Language). Yhdellä dokumentilla voi olla useita tyylisivuja, joten sen ulkoasua voidaan helposti vaihtaa aina tarvittaessa. Näin voidaan dokumentista tehdä nopeasti esimerkiksi suurempia kirjasimia käyttävä versio (Bedunah, 2000).

XML-dokumenttien rakenne

Seuraavaksi käymme läpi XML-dokumenttien rakenteen. Käsittelemme erityisesti niitä XML:n piirteitä, jotka esiintyvät luvussa 4 kuvattavan SABLE:n dokumenttityyppimäärittelyssä.

XML-dokumentti alkaa yleensä *määrittelyllä* (XML-declaration), joka määrittelee, mihin XML-versioon dokumentti perustuu (Light, 1997):

```
<?XML version="1.0" ?>
```

Lisäksi XML-määrittelyyn voi kuulua ohjeita mm. siitä, täytyykö dokumenttityyppimäärittelyä käsitellä kokonaan vai osittain.

Seuraavaksi XML-dokumentissa voi seurata dokumenttityyppimäärittely, joka kuvaa dokumentin ulkoiset ja sisäiset entiteetit sekä sellaiset lähteet, joita varten voidaan tarvita apusovelluksia (Light, 1997). Lisäksi dokumenttityyppimäärittelyyn voidaan liittää ohjeita XML-dokumenttia käsitteleviä ohjelmia varten. Dokumenttityyppimäärittelyyn sisältyvät säännöt voivat sijaita myös erillisessä tiedostossa, joka nimetään ennen varsinaisen dokumentin alkamista:

```
<?XML version="1.0" ?><!DOCTYPE radiokuunnelma  
SYSTEM "kuunnelma.dtd">
```

Varsinainen XML-dokumentti koostuu yhdestä pääelementistä. Tämä pääelementti voi sisältää muita elementtejä, mutta ei itse voi olla osa mitään muuta ele-

menttiä. Kaikilla elementeillä on selkeät rajat, jotka merkitään alku- ja lopputunnisteilla. Kumpikin näistä tunnisteista sisältää lisäksi elementin nimen. Ns. *tyhjät elementit* eivät sisällä mitään tekstiä, joten niiden alku- ja lopputunniste merkitään yhteen:

```
<tauco/>
```

Tyhjät elementit määritellään dokumenttityypimäärittelyssä seuraavasti:

```
<!ELEMENT tauco EMPTY>
```

Jos elementti ei ole tyhjä, on dokumenttityypimäärittelyssä kerrottava, mitä elementti voi sisältää:

```
<!ELEMENT nimi (#PCDATA )>
```

Nimi-elementti voi tämän määrittelyn mukaisesti sisältää mitä tahansa merkkijonodataa eli dokumentin varsinaista tekstiä, mutta ei muita elementtejä.

Attribuuttien avulla voidaan lisätä elementteihin lisäinformaatiota. Attribuutit esiintyvät elementin alkutunnisteen sisällä. XML-dokumenteissa kaikkien elementtien attribuuttiarvojen tulee esiintyä lainausmerkkien sisällä:

```
<KUUNNELMA LAJI="satiiri" PITUUS="45">
```

Kaikki attribuutit ja niiden mahdolliset arvot löytyvät dokumenttityypimäärittelystä:

```
<!ATTLIST kuunnelma LAJI (satiiri | komedia | draama  
| historiallinen | yleinen) "yleinen"  
PITUUS CDATA #IMPLIED>
```

Kuunnelma-elementille voidaan siis dokumentissa määritellä attribuutit LAJI ja PITUUS. LAJI on *lueteltu attribuutti*: se voi saada arvokseen joko ”satiiri”, ”komedia”, ”draama”, ”historiallinen” tai ”yleinen”. Jos LAJI-attribuuttia ei määritellä, se saa oletusarvokseen ”yleinen”. PITUUS-attribuutti on sen sijaan merkkijonotyyppiä (CDATA), ja se voi jäädä määrittelemättä itse dokumentissa. Mikäli kuunnelman pituuden määrittäminen halutaan tehdä pakolliseksi, kirjoitetaan PITUUS-attribuutin kohdalle määritteen #IMPLIED sijasta määrite #REQUIRED.

Dokumenttityypimäärittelyssä voi esiintyä myös ns. *parametrientiteettejä* (parameter entity):

```
<!ENTITY %kohtausosat " vuorosana | tauko | kertoja  
| aani ">
```

Parametrientiteettejä voidaan käyttää makrojen tavoin, jolloin vältytään pitkien tekstipätkien toistuvasta kirjoittamisesta:

```
<!ELEMENT kohtaus (%kohtausosat)*>
```

Kohtaus-elementti voi siis koostua useista vuorosanoista, tauoista, kertojan osuuksista ja muista äänistä missä tahansa järjestyksessä. Esiintymäkertaoperaattori * tarkoittaa, että kohtausosa-entiteetti esiintyy kohtaus-elementissä joko yhden tai useamman kerran tai sitten ei ollenkaan. Vastaavasti käytetään kysymysmerkkiä (?) ja plusmerkkiä (+) osoittamaan, että esiintymiä on joko yksi tai ei yhtään (kysymysmerkki) tai että esiintymiä on yksi tai useampi (plusmerkki).

Itse XML-dokumentissa käytetään sisäisiä ja ulkoisia entiteettejä. *Sisäisten entiteettien* arvo annetaan niiden määrittelyssä:

```
<!ENTITY paahlo "Aino Berttuli Celia Delia Eeva  
Fiona Gerttuli">
```

Tämän jälkeen määriteltyä entiteettiä voidaan käyttää dokumentissa vaikkapa seuraavasti:

```
<vuorosana>Minä olen &paahlo; .</vuorosana>
```

Ulkoisten entiteettien määrittelyssä voidaan viitata johonkin paikkaan, jossa haluttu tieto sijaitsee:

```
<!ENTITY kohtaus1 SYSTEM "kohtaus1.xml">
```

Tämä yhdistää nimen ”kohtaus1” URL-osoitteeseen ”kohtaus1.xml”. Tästä osoitteesta löytyy tiedosto, josta XML-jäsenin saa selville entiteetin sisällön. Jos kyseessä on binäärientiteetti tarvitaan notaatiota, joka kuvaa lähteen tyyppiä:

```
<!ENTITY haukku SYSTEM "haukku.wav" NDATA WAV>
```

Notaatiosta käy ilmi, että ”haukku.wav” on WAV-tyyppinen äänitiedosto.

Ulkoinen entiteetti voidaan määritellä myös seuraavasti (Light, 1997):

```
<!ENTITY image1 PUBLIC "-//RBL//NONSGML Illustration  
1//EN" "images/illus1.gif" NDATA gif>
```

PUBLIC-määrite mahdollistaa entiteetin kuvaamisen. Jos kuvattu entiteetti löytyy paikallisesti saatavilla olevista lähteistä, ei kyseisestä entiteetistä tarvita uutta kopiota.

XML-dokumenttien konvertointi

Konvertointi tarkoittaa dokumenttien muuntamista yhdestä muodosta toiseen. XML-dokumenttien yhteydessä konvertointi voi tarkoittaa kolmea erilaista muunnosprosessia (Light, 1997; Immosen (1999) mukaan Travis & Waldt, 1995):

ei-XML-dokumentin muuntamista XML-muotoon (up-conversion), dokumentin muuntamista yhdestä XML-formaatista toiseen (transformation) tai XML-dokumentin muuntamista tulostuskelpoiseksi dokumentiksi (down conversion). Käytämme jatkossa Travin ja Waldtin (1995) mallin mukaisesti nimitystä *konvertointi* ensiksi mainitussa tapauksessa ja nimitystä *muunnos* silloin, kun lähdedokumenttina on XML-dokumentti.

Puhtaasta tekstistä koostuvan dokumentin konvertoiminen XML-muotoon vaatii yleensä paljon työtä. Ongelmana on se, että lähdedokumentti sisältää tyypillisesti vähemmän informaatiota kuin sen XML-version tulisi sisältää: esimerkiksi kursivoitu teksti voi tarkoittaa vieraskielistä sanaa, korostusta, teknistä termiä tai ironiaa (Light, 1997).

Konvertointi vaatii huolellista dokumentin analysointia sen sisältämien rakenteiden määrittelemiseksi (Immosen (1999) mukaan Travis & Waldt, 1995). Konvertointi voi perustua dokumentin ulkoasuun (format tagging), rakenteeseen (structure tagging) tai sisältöön (content tagging). Konvertointi voi olla manuaalista, jolloin lähdeaineisto käydään läpi käsin esimerkiksi tekstieditoria käyttäen, tai ohjelmallista, jolloin lähdeaineiston muuntamisen XML-muotoon suorittaa erityinen konversio-ohjelma. Konversio-ohjelma pystyy usein tunnistamaan dokumentin rakenneosat niiden ulkoasun perusteella, jonka lisäksi konvertointi voi perustua myös tekstissä esiintyviin avainsanoihin.

Konvertointiprosessi sisältää useita vaiheita (Immosen (1999) mukaan Travis & Waldt, 1995). Ensin suoritetaan varsinainen konvertointi, jonka jälkeen saatu dokumentti tarkistetaan XML-jäsentimen avulla ja löydetyt virheet korjataan. Lopuksi seuraa varmistusvaihe, jonka tarkoituksena on varmistaa tiedon laadukkuus ja semanttinen oikeellisuus. Konvertointiprosessi sisältää yleensä sekä ohjelmallista että manuaalista konvertointia.

Muunnosohjelmien teko on hieman helpompaa, koska lähdedokumenttina on rakenteinen dokumentti. Muunnokset yhdestä XML-tyylisestä formaatista toiseen (esimerkiksi XML:stä HTML:ään) voidaan tehdä skriptillä, joka sisältää ohjeet

siitä, mitä kullekin elementtityypille tehdään muunnoksessa. Tulostuskelpoisen dokumentin saamiseksi XML-dokumentista voidaan puolestaan käyttää hyväksi tyyllisivuja (Light, 1997).

4 SABLE

Seuraavaksi perehdymme tarkemmin SABLE:en ja sen ominaisuuksiin. Aluksi käymme lyhyesti läpi SABLE:n edeltäjät, jotka ovat ratkaisevasti vaikuttaneet tämän merkkauškielen nykyiseen muotoon.

4.1 Puhesynteesiä varten kehitellyt merkkauskielet

Eräs ensimmäisistä puhesynteesiä varten kehitellyistä merkkauškielistä on *SSML* (Speech Synthesis Markup Language), jonka kehitti Paul Taylor vuonna 1992 (Taylor & Isard, 1997). *SSML* on *SGML*-pohjainen ja sen tunnisteeet ovat ”korkeamman tason” tunnisteeite, joilla ei voida kontrolloida puhesyntetisaattoreita kovinkaan tarkasti, mutta joita toisaalta pystyvät helposti ymmärtämään myös sellaiset käyttäjät, joilla ei ole kielitieteen tuntemusta. Ajatuksena *SSML*:ssä oli luoda synteettistä puhetta, joka ei välttämättä kuulostaisi identtisiltä eri järjestelmissä, mutta joka antaisi kuitenkin aina saman perusvaikutelman.

SSML:ssä on käytössä tunnisteeet *PHRASE*, *EMPH*, *DEFINE*, *SRC* ja *LANGUAGE*, joilla voidaan kontrolloida lausepainotusta, tiettyjen sanojen ääntämistä, puheen väliin soitettavia äänitiedostoja ja käytössä olevaa kieltä. *PHRASE*-tunniste sisältää lisäksi kaksi attribuuttia, joiden avulla pystytään määrittelemään intonaatiokontuuria.

STML (Spoken Text Markup Language) saatiin yhdistämällä *SSML* Bell Labsin *TP*-järjestelmässä käytössä olleeseen kattavaan tunnisteejoukkoon (Sproat & al, 1997). Uusina tunnisteeina tulivat mukaan esimerkiksi *SPEAKER*, *GENRE* ja

DIV, joilla voidaan nimetä käytettävä ääni, tekstin tyyllilaji ja tyyllilajin sisäiset jaottelut.

Sun Microsystems on kehittänyt oman vastaavan merkkaukielensä *JSML:n* (Java Speech Markup Language), joka pohjautuu XML:ään (Java Speech Markup Language Specification, 2000). JSML mahdollistaa entistä tarkemman prosodian ohjauksen, sillä siihen otettiin mukaan PROS-tunniste, jonka attribuuteilla voidaan määritellä puhenopeutta, äänenvoimakkuutta, sävelkorkeutta ja sävelalaa. Tekstiin voidaan lisätä myös *merkkejä* (markers) ja puhesyntetisaattorikohtaisia ohjeita.

4.2 SABLE

SABLE on yhä kehitysvaiheessa oleva merkkaukieli, josta toivotaan standardia puhesynteesissä käytettävälle merkkaukselle (*SABLE: A Synthesis Markup Language*, 2000). Tarkoituksena on kehittää merkkaukieli, joka olisi helppokäyttöinen, laajennettava ja sopiva mahdollisimman monen eri kielen merkkaukseen. Näin mahdollistettaisiin myös erilaisten puhesynteesissä tarvittavien työkalujen kuten jäsentimien ja tekstiä SABLE:ksi konvertoivien ohjelmien kehittäminen.

SABLE on kolmen aikaisemmin kehitetyn merkkaukielen, SSML:n, STML:n ja JSML:n, yhdistelmä (*SABLE: A Synthesis Markup Language*, 2000). Se sisältää sekä tekstin rakennetta kuvaavia että puhetta ohjaavia tunnisteita. Seuraavassa kuvaamme lyhyesti SABLE:n tämän hetkisen version eli 1.0:n tunnisteet. Kaikki esimerkit ovat kirjoittajan omia, ellei toisin ole mainittu.

Tekstin kuvaus (text description)

Tekstin kuvaukseen tarkoitettut tunnisteet ovat tunnisteita, jotka antavat tietoa tekstin rakenteesta kuten kappaleista, lauseista, taulukoista ja otsikoista (Sproat & al, 1997).

DIV

DIV-tunniste luokittelee merkityn alueen tietyn tyyppiseksi. Tyyppi määritellään

attribuutilla TYPE, joka tämän hetkisessä versiossa voi saada arvot ”sentence” ja ”paragraph”. DIV-tunnisteen avulla SABLE-dokumenttiin voidaan siis merkitä virkkeet ja kappaleet:

```
<DIV TYPE="paragraph"><DIV TYPE="sentence">
Myydään mustia labradorinnoutajan pentuja.</DIV>
<DIV TYPE="sentence">Puh. 123123 </DIV></DIV>
```

DIV-tunnisteet voivat auttaa TP-järjestelmää tekemään rakenneanalyysia luettavasta tekstistä (Walkers & Hunt, 2000).

SAYAS

Tekstin normalisoinnin yhteydessä TP-järjestelmän on ratkaistava oikeaan lukemistapaan liittyvät monitulkintaisuudet. Esimerkiksi ”123123” voi asiayhteydestä riippuen olla perusluku tai puhelinnumero, joka taas vaikuttaa siihen, miten se luetaan ääneen (”satakaksikymmentäkolmetuhatta satakaksikymmentäkolme” tai ”yksi kaksi kolme, yksi kaksi kolme”).

SAYAS-tunnisteen avulla voidaan lisätä dokumenttiin oikean lukemistavan löytämistä helpottavaa tietoa. Kyseisen tunnisteen attribuutti MODE voi saada arvot ”literal” (tavaus), ”date” (päivämäärä), ”time” (kellonaika), ”net” (www-osoite tai sähköpostiosoite), ”name” (erisnimi), ”currency” (valuutta), ”phone” (puhelinnumero), ”postal” (postiosoite), ”math” (matemaattinen ilmaisu), ”fraction” (murtoosa), ”measure” (mitta), ”ordinal” (järjestysluku) tai ”cardinal” (perusluku). Attribuutilla MODETYPE voidaan puolestaan tarkentaa MODE-attribuutin arvoja ”date”, ”time” ja ”net”. Mahdollisia tarkennuksia ovat erilaisten päivämäärä- ja kellonaikamuotojen lisäksi ”email” ja ”url”.

Edellisen esimerkkiin voidaan siis lisätä seuraavanlainen SAYAS-tunniste, joka selventää TP-järjestelmälle, että ”123123” on puhelinnumero:

```
<DIV TYPE="paragraph"><DIV TYPE="sentence">Myydään
```



```
mustia labradorinnoutajan pentuja.</DIV>  
<DIV TYPE="sentence">Puh.<SAYASMODE="phone">123123</SAYAS>  
</DIV></DIV>
```

Puheen ohjaus (speaker directives)

Puheen ohjaukseen tarkoitetut tunnisteet eivät liity suoraan tekstin rakenteeseen. Niillä vaikutetaan kuitenkin siihen, miten järjestelmä tulkitsee tekstin: minkälainen puheääni valitaan, mikä on tietyn lauseen sävelala jne (Sproat & al, 1997). Tunnisteita EMPH, BREAK, PITCH, RATE ja VOLUME voidaan käyttää ohjaamaan TP-järjestelmää oikeiden prosodisten parametrien löytämisessä (vrt. W3C:n määritelmän (Walkers & Hunt, 2000) EMPHASIS, BREAK ja PROSODY).

EMPH

Mikäli halutaan varmistua siitä, että lausepainot sattuvat kohdalleen, voidaan painotettava sana merkitä EMPH-tunnisteella. Sen attribuutilla LEVEL määritellään korostuksen taso joko sanallisesti ("strong /moderate /none /reduced") tai käyttämällä positiivista kokonais- tai desimaalilukua. Oletusarvona on "moderate". Kuvan 2 esimerkkilause kontrastiivisesta painosta voitaisiin merkata näin:

```
<DIV TYPE="sentence">Tuo <EMPH>kissa</EMPH>  
käyttäytyy kummallisesti.</DIV>
```

BREAK

Sopiviin paikkoihin lisätyt tauot tekevät puheesta selkeämpää ja helpommin ymmärrettävää. BREAK-tunnisteella voidaan lisätä dokumenttiin virkkeen sisäinen, prosodinen tauko (vrt. DIV, jonka avulla määritellään virkkeiden ja kappaleen väliset tauot). Tauon kesto määritellään attribuutin LEVEL avulla joko sanallisesti ("large /medium /small /none") tai kokonais- tai desimaaliluvulla. Negatiivisia arvoja voidaan käyttää osoittamaan sanan sisällä esiintyviä, lyhyempiä taukoja. Vaihtoehtoisesti tauon pituus voidaan määritellä myös millisekunneissa MSEC-attribuutilla. Lisäksi voidaan kiinnittää halutunlainen intonaatiokonttuuri taukoa

edeltävään lauseeseen asettamalla TYPE-attribuutin arvoksi joko "?" (kysymys), "!" (huudahdus), "." (väittäjä) tai "," (virke jatkuu).

Seuraavassa esimerkissä havainnollistetaan BREAK-tunnisteen käyttöä monisitteisen lauseen disambigoinnissa:

```
<DIV TYPE="sentence">Bon Appétit.Herkkuja mädistä  
<BREAK LEVEL="small"/>ja makeista sitrushedelmistä.  
</DIV>
```

PITCH

Pitch-tunnisteella voidaan määritellä valitun alueen sävelkorkeutta. Attribuutteina ovat RANGE, joka kertoo intonaation vaihteluvälin, sekä BASE ja MIDDLE, joilla voidaan määritellä puhujan F0:n minimi (bottom line) ja vaihteluvälin keski-kohta (middle line). Attribuuttien arvot voidaan antaa joko hertseinä, prosentteina kuvaten laskua tai nousua nykyiseen, tai sanallisesti ("highest /high /medium /low /lowest /default" attribuuteilla BASE ja MIDDLE sekä "largest /large /medium /small /smallest /default" attribuutilla RANGE).

RATE

Puhenopeutta voidaan säädellä RATE-tunnisteella. Sen attribuuttina on SPEED, jonka arvona voi olla joko positiivinen desimaaliluku (sanoja per minuutti), prosenttiluku kuvaten nopeutumista tai hidastumista sen hetkiseen puhenopeuteen nähden tai sanallinen kuvaus asteikolla "fastest /fast /medium /slow /slowest". RATE-tunnisteen avulla voidaan esimerkiksi hidastaa TP- järjestelmän lukemisnopeutta silloin, kun on tulossa jotain tärkeää tai muistiinkirjoitettavaa tietoa:

```
<DIV TYPE="paragraph"><DIV TYPE="sentence">Myydään  
mustia labradorinnoutajan pentuja.</DIV>  
<DIV TYPE="sentence">Puh.<RATE SPEED="-50\%">123123  
</RATE></DIV></DIV>
```

VOLUME

Joissain tapauksissa voi olla tarpeen säätää äänenvoimakkuutta dokumentin eri kohtia luettaessa. Tähän käytetään SABLE-dokumenteissa VOLUME-tunnistetta, jonka attribuutti LEVEL voi saada arvokseen joko ykkösen ja nollan väliltä olevan desimaaliluvun (1 - syntetisaattorin suurin äänenvoimakkuus, 0 - hiljaisuus), prosenttiluvun, joka kuvaa muutosta sen hetkiseen äänenvoimakkuuteen tai kuvaavan termin (”loudest /loud /medium /quiet”). Oletuksena on ”medium”.

AUDIO

AUDIO-tunnisteen avulla voidaan luettavan tekstin joukkoon laittaa soitettavaksi äänitiedostoja. Attribuutilla SRC määritellään halutun dokumentin URL-osoite, MODE-attribuutilla puolestaan voidaan asettaa äänitiedosto soitettavaksi taustalle (”background”) tai keskeyttää tekstin lukeminen äänitiedoston soittamisen ajaksi (”insertion”). Attribuutilla LEVEL määritellään, millä äänenvoimakkuudella äänitiedosto soitetaan. Se saa arvokseen desimaaliluvun ykkösen ja nollan väliltä (1.0 - kuten ympäröivä puhe, 0.0 - hiljaisuus). Äänitiedoston lisääminen luettavaan tekstiin tapahtuu seuraavasti:

```
<DIV TYPE="paragraph"><DIV TYPE="sentence">Myydään  
mustia labradorinnoutajan pentuja.</DIV>  
<AUDIO SRC="haukku.wav" MODE="insertion"  
LEVEL="0.5"/><DIV TYPE="sentence">Puh. 123123  
</DIV></DIV>
```

ENGINE

Dokumenteissa voi joskus olla tarpeen antaa tietyille puhesyntetisaattorille tarkoitettuja ohjeita, jotka muiden syntetisaattoreiden täytyy jättää huomiotta. TP-järjestelmä korvaa ENGINE-tunnisteella merkityn alueen sen DATA-attribuutin arvolla, mikäli syntetisaattorina on attribuutin ID määrittelemä syntetisaattori. Esimerkki:

```
<DIV type="sentence">Tämän tekstin teille lukee
```

```
<ENGINE ID="Festival" DATA="maailman paras
syntetisaattori!">joku muu kuin
Festival-syntetisaattori.</ENGINE></DIV>
```

PRON

PRON-tunnisteen avulla voidaan määrittellä halutunlainen ääntämys merkatun alueen tekstile ja varmistaa näin, että TP-järjestelmä löytää ongelmallisissa tapauksissa oikeat äänteet grafeemi-foneemi-konversion aikana. ORIGIN-attribuutti määrittelee merkatun alueen kielen, ja ääntämisohjeet voidaan puolestaan antaa attribuuteilla IPA (Unicode IPA:n mukainen merkkijono) tai SUB ("foneettinen tavaus" kyseessä olevalla kielellä). Näin voidaan kertoa TP-järjestelmälle, miten jokin sille vieras sana tai lyhenne tulisi ääntää:

```
<DIV TYPE="paragraph"><DIV TYPE="sentence">Myydään
mustia labradorinnoutajan pentuja.</DIV>
<DIV TYPE="sentence"><PRON SUB="puhelin"> Puh.</PRON>
123123</DIV></DIV>
```

LANGUAGE

SABLE-dokumentissa voidaan käyttää useampia eri kieliä tai murteita, jos ne merkataan LANGUAGE-tunnisteella. Attribuutti ID kertoo käytettävän kielen tai murteen:

```
<LANGUAGE ID="Finnish"><DIV TYPE="sentence">Myydään
mustia labradorinnoutajan pentuja.</DIV></LANGUAGE>
<LANGUAGE ID="English"><DIV TYPE="sentence">Black
Labrador retriever puppies for sale</DIV>
</LANGUAGE>
```

SPEAKER

Dokumenttien teksti voidaan lukea käyttäen useita eri puheääniä. SPEAKER-

tunniste asettaa merkätun alueen puhujaksi tietynlaisen puhujan: GENDER-attribuutti määrittelee sukupuolen ("male / female"), AGE iän ("older /middle /younger /teen /child") ja NAME puhujan nimen siinä tapauksessa, että käytetään jotain tiettyä puhesyntetisaattoria. Esimerkiksi kahden eri uutistenlukijan vuoro-sanat voitaisiin merkata seuraavasti:

```
<SPEAKER GENDER="male" AGE="middle">
<DIV TYPE="sentence">Ja uutisista nyt
näkemiin ja hyvää illan jatkoa.</DIV>
</SPEAKER><SPEAKER GENDER="female" AGE="middle">
<DIV TYPE="sentence">Näkemiin.</DIV> </SPEAKER>
```

Muut tunnisteet

Tähän ryhmään kuuluvat tunnisteet, jotka eivät suoranaisesti vaikuta siihen, miten merkattu teksti luetaan.

SABLE

SABLE-tunniste määrittelee dokumentin SABLE-dokumentiksi. Se rajaa dokumentin varsinaisen tekstiosuuden pääelementin, jonka sisällä kaikki muut mahdolliset elementit esiintyvät.

MARKER

Jokainen SABLE:n tunniste voi sisältää MARK-attribuutin, jolla voidaan asettaa merkki haluttuun kohtaan tekstissä. Mikäli halutussa kohdassa ei ole tunnistetta, jonka MARK-attribuuttia voitaisiin käyttää, käytetään merkin asettamiseen MARKER-tunnistetta.

4.3 SABLE:n käyttäminen synteesijärjestelmissä

SABLE-merkkaus on kehitelty lähinnä merkkauksen automaattista generoimista silmällä pitäen (Sproat & al, 2000), mutta sen tunnisteita voidaan melko helposti

lisätä tekstiin myös käsin (Wouters & al, 1999). Se, millä tavalla TP-järjestelmän syötedokumentti generoidaan, riippuu siitä, kuinka paljon dokumentin tekijällä (eli käännöksen tekevällä sovelluksella, ihmisellä tai molemmilla) on tietoa merkattavan tekstin sisällöstä (Walkers & Hunt, 2000). Jos syötedokumentin tekijällä ei ole tietoa, jolla merkata syötetekstiä, TP-järjestelmän täytyy selvittää puheen tuottamisesta omien analyysiensä perusteella. Toisaalta mikäli syötedokumentti tehdään käsin, se voidaan merkata hyvinkin tarkasti, jolloin varmistetaan siitä, että lopputulos on mahdollisimman samanlainen käytettävästä puhesyntetisaattorista riippumatta.

Jos merkatun tekstin luonti halutaan tehdä ohjelmallisesti, tarvitaan sovelluksia (Sproat & al, 2000). Tällainen sovellus voisi olla esimerkiksi konversio-ohjelma, joka merkkaisi sähköpostista kaikki otsikot, lainaukset ym. SABLE:n tunnisteilla siten, että mikä tahansa SABLE-yhteensopiva TP-järjestelmä voisi lukea kyseisen sähköpostin. Sovelluksia voidaan kirjoittaa myös muuntamaan \LaTeX - ja Ms Word-muotoisia dokumentteja SABLE-muotoon. Mikäli muunnettava dokumentti on jo valmiiksi SGML/XML-tyyppisessä muodossa (esim. HTML), voidaan muunnoksen tekemisessä hyödyntää jo olemassaolevia dokumentin käännösjärjestelmiä.

Jos puhesyntetisaattori ei suoraan tue SABLE:a vaan käyttää omaa merkkaustaan, tarvitaan vielä sovellus, joka muuntaa SABLE-tunnisteet kyseisen syntetisaattorin omiksi tunnisteiksi (Sproat & al, 2000). Tämän voi toteuttaa esimerkiksi XML-jäsentimen avulla tai PERL-kieltä käyttäen. Täydellinen tiedoston muuntaminen esiprosessointivaiheessa on useimmiten riittävä. Jos puhesyntetisaattori ei tue jotain SABLE:n tunnistetta, se jättää sen yleensä huomioimatta. Tästä seuraa suurempia ongelmia vain SPEAKER- ja LANGUAGE-tunnisteiden yhteydessä.

Internet-sivuja varten on suunniteltu tyylisivujen audiovastinetta, jota kutsutaan nimellä ACSS (Aural Cascaded Style Sheets) (Sproat & Raman, 2000). Kun tavallisissa tyylisivuissa voidaan määritellä esimerkiksi erilaisten otsikoiden graafinen ulkoasu, voidaan ACSS:n avulla vastaavasti määritellä elementeille ääniominaisuudet. Tämän jälkeen ACSS:n määrittelyt voidaan toteuttaa syntetisaatto-

ririippumattomalla tavalla käyttäen SABLE:a ja saatu teksti voidaan sitten lukea käyttäjälle tämän haluamalla TP-järjestelmällä. ACSS:n kaltaisia tyylisivuja voitaisiin hyödyntää myös muissa sovelluksissa, joilla on jo ennakolta tietoa puhuttavan tekstin sisällöstä (SABLE: A Synthesis Markup Language, 2000). Tällöin sovellus käyttäisi tyylisivuja tuottaakseen muunnoksen jo olemassa olevista XML-dokumenteista SABLE-muotoon.

4.4 SABLE:n arviointi

SABLE perustuu olemassa olevaan ja vakiintuneeseen standardiin, XML:ään. Niinpä valmiita työkaluja tekstin muokkaukseen on jo olemassa, ja lähtökohdat merkkaukielen kehittymiselle ovat hyvät. SABLE sisältää myös kohtalaisen monipuolisen tunnistejoukon, jolla voidaan parantaa puhesyntetisaattorin tuottaman puheen laatua - edellyttäen tietenkin, että puhesyntetisaattori osaa noudattaa ja tarvittaessa toteuttaa tarpeeksi hyvin SABLE-tunnisteiden ohjeita.

SABLE tarjoaa mahdollisuuden sekä hieman abstraktimpaan että hyvinkin tarkkaan syntetisaattorin ohjaukseen: esimerkiksi korostus voidaan haluttaessa merkitä joko pelkästään EMPH-tunnisteella ja jättää näin varsinainen toteutus kokonaan puhesyntetisaattorin huoleksi tai vaihtoehtoisesti toteuttaa itse tarkasti esimerkiksi puhenopeutta ja sävelkorkeutta muuttamalla. Lisäksi SABLE:ssa on mahdollisuus antaa attribuuttien arvot joko sanallisesti tai tarkkoina lukuina (Sproat & al, 2000).

Kehittäjät voivat lisätä SABLE:en omia tunnisteitaan, attribuuttejaan ja attribuuttien mahdollisia arvoja (SABLE :A Synthesis Markup Language, 2000). Mikäli laajennukset yleistyvät ja osoittautuvat käyttökelpoisiksi, ne voidaan tarvittaessa sisällyttää osaksi SABLE:n standardimäärittystä.

SABLE:n huonoja puolia on ainakin se, että se ei sisällä kovinkaan paljon mahdollisuuksia varsinaiseen tekstin kuvaukseen. Toisaalta tekstin kuvaukseen tarvittavien tunnisteiden määrä vaihtelee varsin paljon dokumenttityypistä riippuen, jo-

ten yleispätevän tunnistejoukon määrittelemine on hankalaa. Tällöin tarvittavien tekstinkuvauselementtien määrittely jää merkkauškielen käyttäjän tehtäväksi. Yhtenä vaihtoehtona olisi määrittellä dokumenteille kokonaan oma rakenteensa ja elementtinsä, jolloin nämä tekstin rakennetta kuvaavat elementit voitaisiin myöhemässä vaiheessa korvata halutuilla puhetta ohjaavilla SABLE-elementeillä.

Kritiikkiä SABLE on saanut myös siitä, että sen käyttämät ”matalan tason” tunnisteet estävät kehittynyttä puhesyntetisaattoria päättämästä itse siitä, kuinka toteuttaa parhaiten esimerkiksi korostus (Monaghan, 1999). Mikäli tällaista vaihtoehtoa ei kuitenkaan olisi, tulisi käytössä olla puhesyntetisaattori, jonka tiedetään olevan tarpeeksi kehittynyt voidakseen itse ratkaista tällaiset tapaukset. Tässä mielessä SABLE on pitkälti samassa tilanteessa kuin HTML: merkkauškielten alkuperäinen idea dokumentin loogisen ja fyysisen rakenteen erottamisesta ei enää päde, kun merkkaukseen sekoitetaan myös tunnisteita, jotka suoraan kontrolloivat tekstin toteutumista selaajasta tai syntetisaattorista riippumattomasti. Toisaalta joskus on vaikeaa vetää rajaa siihen, mitkä äänidokumentin ominaisuudet ovat loppujen lopuksi osa itse dokumenttia ja mitkä osa sen toteutusta (Sproat & al, 1997). Esimerkiksi puheäänänen valinta voi jossain tapauksessa jäädä TP-järjestelmän valittavaksi, kun se taas jossain toisessa tapauksessa voi olla hyvinkin olennainen osa dokumenttia (draamatekstin eri roolien vuorosanat yms).

Vaihtoehdoksi matalan tason tunnisteille on ehdotettu merkkausta, joka keskittyisi kuvaamaan tekstistä puuttuvaa diskurssi-informaatiota: uutta ja vanhaa informaatiota lauseissa, vastakohtaisuuksia jne. (Monaghan, 1999). Tällainen informaatio olisi esimerkiksi prosodian tuottamisen kannalta äärimmäisen hyödyllistä, mutta sitä on hankala löytää tekstistä pelkän automaattisen analyysin avulla. Kaikkien tällaisten kohtien merkkaaminen tekstiin käsin olisi puolestaan varsin hidasta.

Prosodian tarkempi ohjaus vaatisi SABLE:n tunnisteisiin lisäattribuutteja, kuten PITCH- ja VOLUME-tunnisteisiin ehdotetut CONTOUR-attribuutit, joiden avulla voitaisiin tarkasti määrittellä haluttu intonaatiokontuuri (Wouters & al, 1999). Segmenttien kestojen määrittelyt toisivat puolestaan lisää mahdollisuuksia esimerkiksi korostuksen toteuttamiseen.

W3C:n (World Wide Web Consortium) määrittelemä puhesynteesiin soveltuva merkkaukieli (Walkers & Hunt, 2000) on pitkälti SABLE:n kaltainen. Siinä on kuitenkin ehdotettu joitakin uusia piirteitä kuten akronyymien oikean ääntämisen mahdollistavaa attribuuttia SAYAS-tunnisteeseen sekä VARIANT-attribuuttia, jolla voitaisiin määritellä toinen, käytetyn äänen ominaisuudet omaava, mutta erilainen ääni. Määrittelyyn on otettu mukaan myös jo edellä mainitut CONTOUR-attribuutti ja prosodiaa määrittelevään tunnisteeseen lisättävä DURATION-attribuutti, jolla voidaan määritellä tietyn elementin sisällön lukemiseen kuluva aika.

SABLE:ssa nykyisin käytössä oleva IPA-merkistö on melko vaikeasti ymmärrettävä, eikä niitä löydy kovinkaan monesta kirjasinlajista. Tulevaisuudessa SABLE:en ollaan mahdollisesti lisäämässä tukea myös muille merkistöille.

ACSS:ssä on mahdollista määrittää ääni kuuluvaksi eri suunnista. Tätä ominaisuutta ei ole ainakaan toistaiseksi sisällytetty SABLE:n määrittelyyn. Samoin jatkossa tullaan todennäköisesti tarvitsemaan keinoja merkata tekstiä niin, että voidaan tukea myös audiovisuaalista synteesiä.

Miten paljon puhesynteesin laatua voidaan sitten SABLE:n avulla parantaa? Jonkinlaisen kuvan asiasta saa antamalla TP-järjestelmälle syötteenä saman tekstidokumentin sekä sellaisenaan että SABLE-tunnisteilla merkattuna ja kuuntelemalla lopputuloksen kummassakin tapauksessa. Tätä tutkielmaa varten otin kaksi lyhyttä tekstiä ja tein niistä SABLE-versiot, joihin lisäsin sellaisia tunnisteita, joiden uskoin lukemani perusteella tekevän tekstistä kuultuna selkeämmän ja helpommin ymmärrettävän. Käytin kokeilussani Festival-järjestelmää (Festival Speech Synthesis System, 2001), joka ymmärtää SABLE-merkattuja dokumentteja. Tekstit ja niiden SABLE-merkatut versiot samoin kuin vastaavat äänitiedostot ovat mukana tutkielman liitteinä 2,3 ja 4.

Tekstissä ”labbis.sable” on Festival-järjestelmän lukemana kaksi merkittävää ongelmaa: järjestelmä lukee tekstissä esiintyvän puhelinnumeron tavallisena lukuna ja lisäksi tämä varsin tärkeä osa tekstiä tulee varsin nopeassa tahdissa. Nämä on-

gelmat on ratkaistu tunnisteita käyttävässä versiossa ”labbis2.sable” SAYAS- ja RATE-tunnisteilla, joiden ansiosta järjestelmä osaa lukea puhelinnumeron oikein ja hidastaa vielä puhenopeutta niin, että kuulija kuulee varmasti puhelinnumeron ja saa sen tarvittaessa kirjoitettua ylös. Lisäksi käytin ENGINE-tunnistetta saadakseni järjestelmän lausumaan lyhenteen ”Tel.” sijasta ”Telephone number is”.

Hieman pitempi tekstitiedosto ”job.sable” on ilman tunnisteita luettuna lähinnä kaoottisen kuuloisen. Tärkeät tiedot luetaan yhteen putkeen ilman sanottavaa jakottelua, ja luettua on vaikea ymmärtää. Huomattavan avun tähän tuo pelkkä erimittaisten taukojen lisääminen sopiviin kohtiin niin, että toisiinsa liittyvät asiat on erotettu pienellä tauolla ja eri kohtiin kuuluvat asiat hieman isommalla tauolla. Tärkeimpiä kohtia on jälleen korostettu hidastamalla niiden kohdalla puhenopeutta. Samoin puhelinnumero ja sähköpostiosoite on merkattu omin tunnistein.

Tämän pienimuotoisen kokeilun päällimmäiseksi vaikutelmaksi jäi se, että joitain ongelmia on helppo korjata SABLE-tunnisteiden avulla. Jo niinkin yksinkertainen asia kuin taukojen lisäys selkiytti tekstin lukemista huomattavasti. Samoin puhelinnumeroiden ynnä muiden erikoistietojen merkkaminen on hyvin kätevä tapa kertoa puhesynteesijärjestelmälle minkälaisesta tiedosta on kysymys.

Kaikki SABLE:n tunnisteet eivät toimineet Festival-järjestelmässä odotetusti. Esimerkiksi EMPH-tunniste oli käytännössä varsin kelvoton, sillä sen käyttäminen tekstissä sai aikaan hyvin epäluonnollisen kuuloisen ja joskus jopa aiempaa epäselvemmän vaikutelman. Lisäksi SABLE:n rajallisuus tuli nopeasti vastaan: järjestelmän monotoniseen luentaan olisi ollut hyvä saada hieman vaihtelua, mutta PITCH-tunniste amatöörifoneetikon käyttämänä lähinnä vain pahensi tilannetta.

5 Yhteenveto

Luonnolliset kielet koostuvat monesta eri tasosta. Se, miten kirjoitettu teksti reaalistuu puheena, on hyvin monimutkainen prosessi, johon ovat osaltaan vaikut-

tamassa kaikki kielen eri tasot. Tekstistä puheeksi-järjestelmissä tätä prosessia on yritetty hahmottaa tietokoneelle sopivaan muotoon.

Toistaiseksi ei ole vielä pystytty ohjelmoimaan tietokonetta niin, että se pystyisi ”ymmärtämään” tekstejä ihmisen tavoin. TP-järjestelmässä pyritään kuitenkin mahdollisimman ymmärrettävään ja selkeään puheeseen, joten tekstin ne piirteet, jotka vaikuttavat ihmisen puheeseen, olisi välitettävä koneelle jollakin muulla tavalla.

Merkkauskielten avulla tekstiin voidaan upottaa lisäinformaatiota kuten tietoa siitä, millaisesta informaatiosta tekstin kussakin kohdassa on kysymys. Käytettäessä jotain esimerkiksi XML:ään pohjautuvaa merkkauskieltä, saadaan aikaan dokumentti, jota on helppo käsitellä ohjelmallisesti.

TP-järjestelmien kohdalla ongelmana on päättää, millaista tietoa järjestelmä tarvitsee, minkä tason tunnisteita olisi käytettävä ja onko näitä tunnisteita mahdollista lisätä tekstiin automaattisesti. Mitä enemmän tunnisteita generoidaan automaattisesti, sitä nopeammin ja helpommin tekstidokumentti saadaan luettavaan muotoon koneelle. Toisaalta puheen tuottamisen kannalta tärkeimmät seikat saattavat olla sellaisia, että ne on pakko merkata tekstiin käsin.

SABLE on hyvä pohja puhesynteesin merkkauskieleksi, mutta se vaatii vielä kehittämistä ja uusia tunnisteita. Pelkät tunnisteet eivät tietenkään vielä riitä: on kehitettävä myös puhesynteesijärjestelmiä, jotka pystyvät hyödyntämään tarpeeksi hyvin tunnisteiden antamaa informaatiota.

Tutkielmaan sisältyvä kokeilu on pienimuotoinen eikä sinällään kerro kaikkea SABLE:n mahdollisuuksista. Se antaa kuitenkin viitteitä siitä, mitkä asiat ovat SABLE:n avulla helposti korjattavissa (tauotus, erikoistiedot) ja mitkä seikat jäävät vielä toistaiseksi sen ulkopuolelle (kuten kunnollinen prosodia).

Viitteet

Bedunah, J.B. *XML: The Future of the Web*. Internet WWW-sivu, URL:
<http://www.acm.org/crossroads/xrds6-2/future.html>
(14.6.2000).

Bhaskararo, P. "Subphonemic Segment Inventories for Concatenative Speech Synthesis". Teoksessa Keller, E.(toim.) *Fundamentals of Speech Synthesis and Speech Recognition*. John Wiley & Sons Ltd, Chichester, 1994.

Festival Speech Synthesis System. Internet WWW-sivu, URL:
<http://www.cstr.ed.ac.uk/projects/festival/> (22.5.2001).

Immonen, J. *Dokumenttien konvertointi SGML-muotoon*. Pro gradu-tutkielma, Joensuun yliopisto, Tietojenkäsittelytieteen laitos, 1999.

Java Speech Markup Language Specification. Internet WWW-sivu, URL:
<http://java.sun.com/products/java-media/speech/forDevelopers/JSML/JSML.html> (9.6.2000).

Karlsson, F. *Yleinen kielitiede*. Yliopistopaino, Helsinki, 1998.

Lemmetty, S. *Review of Speech Synthesis Technology*. Diplomityö, Teknillinen korkeakoulu, Helsinki, 1999.

Light, R. *Presenting XML*. Sams.net Publishing, Indianapolis, 1997.

Monaghan, A. *State-of-the-Art Summary of European Synthetic Prosody R&D*, 1999. Internet WWW-sivu, URL:
http://www.unil.ch/imm/docs/LAIP/COST_258/Scientific/SOAPs/soap__monaghan.htm (27.10.2000).

O'Grady, W. "Semantics: the analysis of meaning". O'Grady, W. & Dobrovolsky, M. & Katamba, F. (toim.) *Contemporary Linguistics - An Introduction*. Longman, Lontoo, 1997.

Pfister, B. & Traber, C. "Text-to-Speech Synthesis: An introduction and a Case Study". Keller, E.(toim.) *Fundamentals of Speech Synthesis and Speech Recognition*. John Wiley & Sons Ltd, Chichester, 1994.

SABLE: A Synthesis Markup Language (version 1.0). Internet WWW-sivu, URL: <http://www1.bell-labs.com/project/tts/sable.html>(12.6.2000).

Salminen, A. *Rakenteisen tekstin hallinta*. Jyväskylän yliopisto, Tietojenkäsittelytieteen laitos, opetusmoniste, 1992.

Sproat, R., Hunt, A., Ostendorf, M., Taylor, P., Black, A., Lenzo, K. & Edgington, M. *SABLE: A Standard for TTS markup*. Internet WWW-sivu, URL: <http://www.research.att.com/~rws/SABPAP/sabpap.htm> (22.5.2001).

Sproat, R. & Raman, T.V. *SABLE: an XML-based Aural Display List for the WWW*. Internet WWW-sivu, URL: <http://www.bell-labs.com/project/tts/csssable.html> (12.6.2000).

Sproat, R., Taylor, P., Tanenblatt, M. & Isard, A. "A Markup Language for Text-to-Speech Synthesis". *Proceedings of EUROSPEECH 97*, Rhodes, Greece, 1997.

Taylor, P. & Isard, A. "SSML: A Speech Synthesis Markup Language". *Speech Communication*, 21(1-2), ss. 123-133, 1997.

Travis, B.E. & Waldt, D.C. *The SGML Implementation Guide. A Blueprint for SGML Migration*. Springer, Berliini, 1995.

Walkers, M.R. & Hunt, A. *Speech Synthesis Markup Language Specification for the Speech Interface Framework*. W3C Working Draft 08 August 2000. <http://www.w3.org/TR/2000/WD-speech-synthesis-20000808> (7.9.2000).

Wiik, K. *Fonetiikan perusteet*. WSOY, Juva, 1981.

Wouters, J., Rundle, B. & Macon, M.W. "Authoring Tools for Speech Synthesis

Using the Sable Markup Standard”. *Proceedings of EUROSPEECH*, Budapest, Hungary, vol 2, ss. 963-966, 1999.

Liite 1: SABLE:n dokumenttityyppimäärittely

```
<!-- <!DOCTYPE sable SYSTEM "Sable.v0_2.dtd" [  
<!  
<!--  
<!-- Centre for Speech Technology Research  
<!-- University of Edinburgh, UK  
<!-- Copyright (c) 1998  
<!-- All Rights Reserved.  
<!--  
<!-- Permission is hereby granted, free of charge, to  
<!-- use and distribute this software and its  
<!-- documentation without restriction, including  
<!-- without limitation the rights to use, copy,  
<!-- modify, merge, publish, distribute, sublicense,  
<!-- and/or sell copies of this work, and to  
<!-- permit persons to whom this work is furnished to  
<!-- do so, subject to the following conditions:  
<!-- 1. The code must retain the above copyright  
<!-- notice, this list of conditions and the  
<!-- following disclaimer.  
<!-- 2. Any modifications must be clearly marked as  
<!-- such.  
<!-- 3. Original authors' names are not deleted.  
<!-- 4. The authors' names are not used to endorse or  
<!-- promote products derived from this software  
<!-- without specific prior written permission.  
<!-- THE UNIVERSITY OF EDINBURGH AND THE CONTRIBUTORS  
<!-- TO THIS WORK DISCLAIM ALL WARRANTIES WITH REGARD  
<!-- TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES
```

```

<!-- OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
<!-- THE UNIVERSITY OF EDINBURGH NOR THE CONTRIBUTORS
<!-- BE LIABLE FOR ANY SPECIAL, INDIRECT OR
<!-- CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER
<!-- RESULTING FROM LOSS OF USE, DATA OR PROFITS,
<!-- WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
<!-- OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
<!-- CONNECTION WITH THE USE OR PERFORMANCE OF THIS
<!-- SOFTWARE.
<!--
<!--
<!-- An early draft of SABLE 0.1 (Jan 8th 1998)
<!--      (Alan W Black awb@cstr.ed.ac.uk)
<!-- This not an official DTD just one that allows
<!-- a mock up of the
<!-- system for Festival-1.2.1 and later
<!--
<!-- Modified to be XML compliant      April 2nd 1998
<!--
<!-- Modified for SABLE 0.2 (August 15, 1998)
<!--      (Richard Sproat rws@research.bell-labs.com)
<!--
<!-- Note that in order to use this as both an XML and
<!-- SGML DTD, we need to declare OMITTAG NO in the
<!-- accompanying sable.decl for SGML.This means that
<!-- end tags are REQUIRED for a sable document to be
<!-- conformant, even when viewed as an SGML document.

<!ENTITY %baseelements "EMPH |
    REAK |
    ITCH |

```



```

        RATE |
        VOLUME |
        AUDIO |
        ENGINE |
        MARKER |
        PRON |
        SAYAS |
        LANGUAGE |
        SPEAKER |

DIV">

<!ELEMENT SABLE (#PCDATA| %baseelements; )*>

<!-- The non-hierarchical ones -->
<!ELEMENT BREAK EMPTY >
<!ATTLIST BREAK
        LEVEL (large|medium|small|none|NUMBER)
        "medium"
TYPE (quest|excl|period|comma) #IMPLIED
MSEC CDATA #IMPLIED
MARK CDATA #IMPLIED>
<!ELEMENT AUDIO EMPTY >
<!ATTLIST AUDIO
        SRC CDATA #IMPLIED
        MODE (background|insertion) "insertion"
LEVEL CDATA #IMPLIED
MARK CDATA #IMPLIED>
<!ELEMENT MARKER EMPTY >
<!ATTLIST MARKER
        MARK CDATA #IMPLIED>

```

```

<!-- The non-nestable ones -->

<!ELEMENT PRON (#PCDATA) >
<!ATTLIST PRON  IPA CDATA #IMPLIED
                SUB CDATA #IMPLIED
ORIGIN CDATA #IMPLIED
MARK CDATA #IMPLIED>
<!ELEMENT SAYAS (#PCDATA) >
<!ATTLIST SAYAS MODE (literal|date|time|phone|net|
postal|currency|math|fraction|measure|
ordinal|cardinal|name) #REQUIRED
                MODETYPE (DMY|MDY|YMD|YM|MY|MD|HM|HMS|
EMAIL|URL) #IMPLIED
MARK CDATA #IMPLIED>

<!-- The nestable ones -->

<!ELEMENT EMPH ( #PCDATA | %baseelements; )*>
<!ATTLIST EMPH  LEVEL (Strong|Moderate|None|Reduced|
NUMBER) "moderate"
                MARK CDATA #IMPLIED>
<!ELEMENT PITCH ( #PCDATA | %baseelements; )*>
<!ATTLIST PITCH BASE CDATA "0%"
                MIDDLE CDATA "0%"
                RANGE CDATA "0%"
                MARK CDATA #IMPLIED>
<!ELEMENT RATE  ( #PCDATA | %baseelements; )*>
<!ATTLIST RATE  SPEED CDATA "0%"
                MARK CDATA #IMPLIED>
<!ELEMENT VOLUME ( #PCDATA | %baseelements; )*>
<!ATTLIST VOLUME LEVEL CDATA "medium"

```

```

                MARK CDATA #IMPLIED>
<!ELEMENT ENGINE ( #PCDATA | %baseelements; )*>
<!ATTLIST ENGINE ID CDATA #IMPLIED
                DATA CDATA #IMPLIED
                MARK CDATA #IMPLIED>

<!ELEMENT LANGUAGE ( #PCDATA | %baseelements; )*>
<!ATTLIST LANGUAGE ID CDATA #IMPLIED
                CODE CDATA #IMPLIED
                MARK CDATA #IMPLIED>
<!ELEMENT SPEAKER ( #PCDATA | %baseelements; )*>
<!ATTLIST SPEAKER GENDER (male|female) #IMPLIED
                AGE (older|middle|younger|
                teen|child) #IMPLIED
                NAME CDATA #IMPLIED
                MARK CDATA #IMPLIED>

<!ELEMENT DIV ( #PCDATA | %baseelements; )*>
<!ATTLIST DIV TYPE (sentence|paragraph) #IMPLIED
                MARK CDATA #IMPLIED>

<!-- Character entities for latin 1 -->

<!ENTITY % ISolat1 PUBLIC
    "-//SABLE//ENTITIES Added Latin 1 for SABLE//EN"
    "sable-latin.ent" >
%ISolat1;

```

Liite 2: Syntetisoidut tekstit ilman tunnisteita

Tiedosto labbis.sable:

```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE
  speech mark up//EN" "Sable.v0_2.dtd"
[]>

<SABLE>

Black Labrador retriever puppies for sale.
Tel.123123.

</SABLE>
```

Tiedosto job.sable:

```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE
  speech mark up//EN" "Sable.v0_2.dtd"
[]>

<SABLE>

Technical Translators
```

30-Mar-2001

Company: Rescon

Job Title: Technical Translators

Location: Dublin

Job Description: Tech Translators in: Swedish,
Finnish, German, Portuguese, Italian.

Key Skills: Relevant European Language

Salary: Negotiable

Job Type: Contract

Contact Name: Susan

Contact Address:

48 Fitzwilliam Square,

Dublin 2

Telephone Number: 01-2501916

Email Address: susan@rescon.ie

</SABLE>

Liite 3: Syntetisoidut tekstit tunnisteilla merkattuna

Tiedosto labbis2.sable:

```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE
  speech mark up//EN" "Sable.v0_2.dtd"
[]>

<SABLE>

Black Labrador retriever puppies for sale.
<ENGINE ID="festival" DATA="telephone number is">
Tel.
</ENGINE>
<BREAK LEVEL="Large"/>
<RATE SPEED="-40\%">
<SAYAS MODE="phone">123123</SAYAS>.
</RATE>

</SABLE>
```

Tiedosto job2.sable:

```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE
```

speech mark up//EN" "Sable.v0_2.dtd"
[]>

<SABLE>

Technical Translators <BREAK LEVEL="large"/>

30-Mar-2001 <BREAK LEVEL="large"/>

Company <BREAK LEVEL="small"/> Rescon

<BREAK LEVEL="large"/>

Job Title <BREAK LEVEL="small"/> Technical Translators

<BREAK LEVEL="large"/>

Location <BREAK LEVEL="small"/> Dublin

<BREAK LEVEL="large"/>

Job Description <BREAK LEVEL="small"/> Tech

Translators in: <RATE SPEED="-20\%"> Swedish,

Finnish, German, Portuguese, Italian</RATE>

<BREAK LEVEL="large"/>

Key Skills <BREAK LEVEL="small"/> Relevant

European Language

<BREAK LEVEL="large"/>

Salary <BREAK LEVEL="small"/> Negotiable

<BREAK LEVEL="large"/>

Job Type <BREAK LEVEL="small"/> Contract

<BREAK LEVEL="large"/>

Contact Name <BREAK LEVEL="small"/> Susan
<BREAK LEVEL="large"/>

Contact Address <BREAK LEVEL="small"/>
<RATE SPEED="-20\%"> 48 Fitzwilliam Square,
Dublin 2</RATE>
<BREAK LEVEL="large"/>

Telephone Number<BREAK LEVEL="small"/>
<RATE SPEED="-40\%"><SAYAS MODE="phone">
01-2501916</SAYAS></RATE>
<BREAK LEVEL="large"/>

Email Address <BREAK LEVEL="small"/>
<RATE SPEED="-40\%"><SAYAS MODE="net"
MODETYPE="email">susan@rescon.ie</SAYAS>
</RATE>

</SABLE>

Liite 4: Levyke

Tutkielman liitteenä oleva levyke sisältää seuraavat tiedostot:

- job.sable
- job2.sable
- job.wav
- job2.wav
- labbis.sable
- labbis2.sable
- labbis.wav
- labbis2.wav
- tutk.ps