
Johdatus WAP-tekniikkaan ja WML-kieleen

1. painos, PDF-digitaaliversio

Kirjoittanut

Lasse Latva, <llatva@cc.hut.fi>

Copyright © 2000-2001 Lasse Latva.

Tämä julkaisu on ladattu digitaalimuodossa (PDF) tekijän henkilökohtaiselta kotisivulta. Julkaisun edelleenlevittäminen kielletty. Voit tehdä julkaisusta paperitulosteen *omaan henkilökohtaiseen (non-profit) käyttöösi*.

Julkaisun tai sen osien käyttö kaupallisiin tarkoituksiin kaikissa muodoissaan kielletty. Mikäli näet julkaisulla kaupallista käyttöä (ts voitit tavalla tai toisella ansaita julkaisun sisällöllä), ota yhteys tekijään (sähköpostiosoite yllä).

Vaikka teoksen esimerkit ovat testattu toimiviksi ja teoksen sisältö perustuu luotettaviin lähteisiin (mm. tekniset standardit ja spesifikaatiot), ei tekijä ota vastuuta teoksen sisällöstä, teoksen mahdollisesti aiheuttamasta suorasta tai epäsuorasta vahingosta tahi teoksen mahdollisesti sisältämistä virheistä.

Oppaan esimerkkien esittämiseen käytetty Nokia 7110-matkapuhelin on vain esimerkki WAP-päätelaitteesta. Esimerkkien esittämiseen Nokia 7110-matkapuhelimella on saatu Nokia Corporationin lupa.

Nokia on Nokia Corporationin rekisteröimä tavaramerkki. Muut teoksessa esiintyvät yritysten ja/tai tuotteiden nimet saattavat olla muiden tahojen omistamia tavaramerkkejä tai kauppanimiä.

Sisällysluettelo

| | | |
|---------------------|--|----------|
| 1 | Johdanto | 3 |
| 2 | Langaton Internet ja WAP | 4 |
| 2.1 | WAP-protokollat | 4 |
| 2.2 | WAP-arkkitehtuuri ja välityspalvelin | 6 |
| 3 | Wireless Markup Language | 9 |
| 3.1 | WML:n sisältötyypit ja merkit | 10 |
| 3.2 | WML päätelaitteessa | 10 |
| 3.3 | WML-kielen komennot | 11 |
| 3.3.1 | WML-pakka, kortit ja WML-dokumenttien kappalejako | 12 |
| 3.3.2 | WML-pakan head-osio ja metatieto | 13 |
| 3.3.3 | Tekstin muotoilu, korostus ja erikoismerkkien käyttö | 14 |
| 3.3.4 | Template | 15 |
| 3.3.5 | WML ja hyperlinkit | 16 |
| 3.3.6 | Kuvat WML-dokumentissa | 18 |
| 3.3.7 | Taulukot | 19 |
| 3.3.8 | Lomakkeet | 20 |
| 3.3.9 | Tapahtumat (events) | 24 |
| 3.3.10 | Tehtävät (tasks) | 26 |
| 3.3.11 | Ajastin | 28 |
| 3.3.12 | Muuttujat | 29 |
| Liitteet | | |
| Liite 1 | Tavallisimmat WML-käskyt | |
| Liite 2 | Lyhennesanasto | |

1 Johdanto

Samanaikaisesti kun matkapuhelinten fyysinen koko on viimeisen vuosikymmenen aikana pienentynyt räjähdysmäisesti, ovat niiden ominaisuudet laajentuneet ja parantuneet huomattavasti, erityisesti datapalveluiden osalta. Tämä mahdollistaa puhelimelle aivan uudenlaiset käyttömahdollisuudet: matkapuhelimesta on kasvamassa interaktiivinen tietoverkkotyökalu.

Jo muutamia vuosia on ollut mahdollista käyttää kannettavaa tietokonetta yhdessä GSM-puhelimen kanssa Internet-selailuun. Yhdistelmän avulla on ollut mahdollista hoitaa sähköpostia paikasta riippumatta tai käyttää WWW:n (World Wide Web) palveluita myös toimiston ulkopuolella, vain esimerkkejä mainitakseni. WAP-tekniikka (Wireless Application Protocol) mahdollistaa esimerkiksi sähköpostin lukemisen GSM-puhelimen näytöltä ilman lisälaitteita.

WAP tarjoaa mahdollisuuden julkaista hypertekstidokumentteja pienikokoisissakin kannettavissa päätelaitteissa. Läheisesti WWW:ssä käytettävää HTML-kieltä (Hypertext Markup Language) muistuttavan WML-kielen (Wireless Markup Language) avulla laaditut dokumentit voivat sisältää tekstiä ja kuvaa, valintalistoja ja tekstieditoreja. Tulevaisuudessa WAP saattaa merkitä reaaliaikaista videokuvaa tai VoIP-puhehytelyksiä. WAP ei ole laitteistosta riippuvainen, vaan soveltuu PDA-laitteeseen tai palmtop-mikroon siinä missä matkapuhelimeenkin.

WAP-tekniikan edut tulevat esille sellaisissa interaktiivisissa palvelusovelluksissa, joissa päätelaitteen pieni koko korostuu. WAP-tekniikan avulla on teknisesti täysin mahdollista toteuttaa esimerkiksi pizzan tilaaminen kotiovelle tai yrityksen varastotietokannan selailu matkapuhelimen näytöltä.

WAP-tekniikkaa ei voida pitää WWW:n korvaajana, vaan kyse on vaihtoehtoisesta välineestä tuottaa interaktiivisia hypertekstipalveluita pieniin päätelaitteisiin, kuten matkapuhelimiin. WAP-päätelaitteen tieto saattaa kuitenkin olla peräisin samasta tietokannasta kuin vastaava WWW-sivu, tai WML-dokumentti saatetaan tietyissä tapauksissa luoda dynaamisesti jopa vastaavasta WWW-sivusta.

Käsittelen tässä teoksessa WAP-tekniikkaa esitellen perusteet WAP-arkkitehtuurista ja WML-sivunkuvauskielestä. Pyrin havainnollistamaan kielen rakenteita esimerkein, esittelemään WML-kielen mahdollisuuksia sekä opettamaan alkeet WAP-kotisivujen luomiseksi WML-kielellä.

Oppaan toisessa luvussa käsittelen WAP-tekniikkaa yleisesti esitellen muun muassa protokollia ja WAP-arkkitehtuuria. Kolmannessa luvussa paneudun tarkemmin WML-kieleen ja esittelen keskeisimmät WML-kielen komennot (tagit) WML 1.1-standardin mukaan.

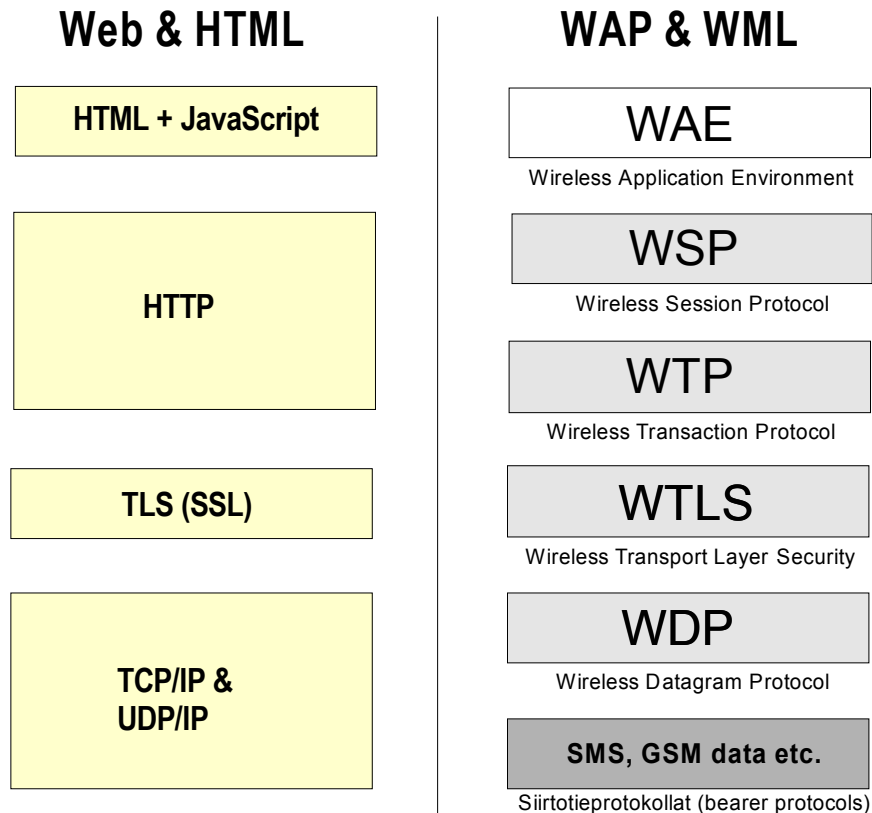
2 Langaton Internet ja WAP

Internetin merkitys maailmanlaajuisena monipalveluverkkona on korostunut Internet-verkon kaupallistumisen ja kasvun myötä erilaisten palvelutyyppien sulautuessa yhteiseen verkkoon. Jo ennestään laaja-alainen Internet mahdollistaa nykyisin sähköpostin ja WWW-sivujen lisäksi myös puheen siirtoa (VoIP) ja langattomia palveluita, kuten WAP-liikennettä, jonka tekniikkaa käsittelemme yleisesti tässä luvussa.

2.1 WAP-protokollat

WAP voidaan jakaa kolmeen osaan, jotka ovat WAE, varsinainen WAP-protokollapino sekä nk. siirtotieprotokollat. WAE:llä tarkoitetaan käyttäjälle näkyvää WAP-sovellusympäristöä. WAP-protokollapino voidaan esittää neljänä kerroksena: WSP, WTP, WTLS ja WDP. Siirtotieprotokollilla tarkoitetaan yhteyskäytäntöjä, joiden avulla WAP-liikennettä voidaan välittää esimerkiksi GSM-verkossa.

Seuraavassa kuviossa esitän WAP-protokollat kerrosmuodossa toiminnoiltaan vastaavien WWW-tekniikan protokollien rinnalla.



Kuvio 2.1 WAP-protokollat

WAE eli *Wireless Application Environment* on yleinen langattoman WAP-teknologian ja olemassa olevan WWW-teknologian yhdistävä langaton sovellusympäristö. WAE-kerrokseen kuuluu muun muassa WAP-selainympäristö. Langattoman maailman WAE vastaa näkyviltä osiltaan WWW-teknikassa HTML-kielen ja JavaScript-sovellusten muodostamaa palvelukokonaisuutta, jota voidaan tarvittaessa täydentää esimerkiksi CGI-rajapinnan tai PHP-kielen avulla. WAE:n sisältö toteutetaan pääosin WML- (Wireless Markup Language) ja WMLScript-kielillä. WAE:hen kuuluu myös edellä mainittujen sisältötyyppien lisäksi muun muassa WTA, Wireless Telephone Application, WAP-teknikkaan pohjautuva puhelinrajapinta.

WSP

WTP

WTLS

WDP

WSP eli *Wireless Session Protocol* tarjoaa sovellusrajapinnan sisältäen kaksi yhteystyyppiä: yhteydellisen ja yhteydettömän muodon. WDP:n päällä toimivaa yhteydetöntä muotoa voidaan käyttää, mikäli sovelluksen ei tarvitse varmistaa tiedon perillemenoa. Hyvä esimerkki tällaisesta sovelluksesta on WWW, jossa kyselyn perillemenoa ei valvota. WTP:n päällä toimivan yhteydellisen muodon tapauksessa WSP huolehtii yhteyden muodostamisesta ja esimerkiksi sisältötyyppien tunnistuksesta.

WTP eli *Wireless Transaction Protocol* on erityisesti pienitehoisille päätelaitteille suunniteltu kyselyprotokolla (Transaction Protocol). WTP-kyselyt voivat olla joko luotettavia yksi- tai kaksisuuntaisia kyselyitä tai epäluotettavia yksisuuntaisia kyselyitä. Jälkimmäisessä tapauksessa kyselyyn ei edellytetä vastausta. WTP sallii myös asynkroniset kyselyt.

WSP

WTP

WTLS

WDP

WSP

WTP

WTLS

WDP

WTLS eli *Wireless Transport Layer Security* on WAP-protokollapinon tietoturvakерros. Läheisesti WWW-teknikan TLS-tietoturvaominaisuuksia (Transport Layer Security, entinen SSL) muistuttavan WTLS:n tarkoituksena on varmistaa tiedon salakirjoitus (kryptaus) käytettävissä olevalla salausmenetelmällä. WTLS ei ole tiedonsiirron kannalta pakollinen protokollakerros, mutta jokin WAE:n sovellus saattaa edellyttää sen olemassaoloa.

WDP eli *Wireless Datagram Protocol* sijaitsee WAP-protokollista alimpana ja hoitaa varsinaisen tiedonsiirron. WDP sopeutuu tarjolla olevaan siirtotiehen ja muodostaa yhteyden siirtotien ja ylemmän tason protokollien välille. Juuri WDP:n kyky sopeutua useisiin siirtoteihin mahdollistaa WAP:n salliman laajan päätelaittevalikoiman.

WSP

WTP

WTLS

WDP

Siirtotieprotokollat (bearer protocols) tarjoavat nimensä mukaisesti mahdollisuuden tiedon siirtoon, fyysisen väylän, jossa WAP-dataa voidaan kuljettaa. WAP on suunniteltu sopeutumaan useisiin erilaisiin päätelaitteisiin, mikä edellyttää useiden eri siirtoteiden tukea.

WAP-siirtotienä voidaan käyttää esimerkiksi SMS-lyhytviestejä tai joko paketti- tai piirikytkentäistä GSM-datayhteyttä.

Koska WDP tarjoaa tiedonsiirtokerroksena rajapinnan siirtotieprotokollan ja muun WAP-pinon välillä, määrittelee WDP käytettävissä olevat siirtotieprotokollat ja tavat, miten tietoa kullakin siirtotiellä siirretään.

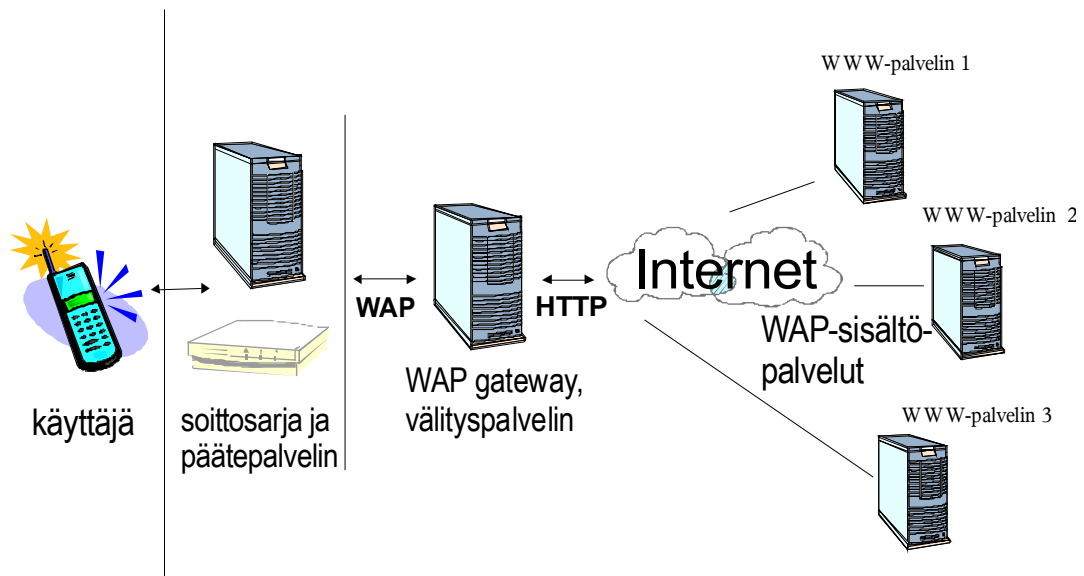
2.2 WAP-arkkitehtuuri ja välityspalvelin

WAP edellyttää välityspalvelimen (WAP-yhdyskäytävä, eli WAP gateway tai WAP proxy) käyttöä. Matkapuhelin käyttää seuraavia (TCP/IP-) portteja muodostaessaan yhteyksiä välityspalvelimen kanssa:

| | | |
|------|-----------------------|----------------------------------|
| 9200 | yhteydetön | connectionless / temporary |
| 9201 | yhteydellinen | connection oriented / continuous |
| 9202 | salattu yhteydetön | connectionless with WTLS |
| 9203 | salattu yhteydellinen | connection oriented with WTLS |

Piirikytkentäinen WAP-yhteys muistuttaa teknisesti hyvin läheisesti valintaisiin modeemiyhteyksiin toteutettuja koti-Internet-yhteyksiä. Otettuaan yhteyden Internet-soittosarjaan ja tämän jälkeen WAP-välityspalvelimeen, on käyttäjällä kaikki Internetin WAP-päätelaitteita varten suunnitellut WML-dokumentit ja muut WAP-palvelut käytettävissään.

WAP-välityspalvelimen eli WAP-yhdyskäytävän voidaan ajatella tavallaan olevan kuin WWW-välimuisti (proxy). Kaikki yhteyden aikana noudettava data kulkee tämän koneen kautta. Tehdäkseen kyselyn WML-sivun noutamiseksi, pyytää käyttäjän matkapuhelimen WAP-selain ensiksi välityspalvelinta noutamaan HTTP-protokollalla kyseisen dokumentin dokumenttia säilyttävältä WWW-palvelimelta. Tämän jälkeen välityspalvelin tarjoaa pyydetyn dokumentin binäärimuotoisena käyttäjän WAP-selaimelle. Mikäli käytettävä sovellus on interaktiivinen, kulkee käyttäjän sovellukselle syöttämä tieto sekin välityspalvelimen kautta aivan kuten päinvastaiseen suuntaan tapahtuvat kyselytkin.



Kuvio 2.2 WAP-arkkitehtuuri

WAP-matkapuhelin ottaa annetun IP-osoitteen perusteella yhteyden välityspalvelimeen. Teknisesti WAP-välityspalvelin on Internetissä sijaitseva palvelin, jossa ajetaan WAP-välityspalvelinohjelmistoa ja jonka IP-osoite kerrotaan matkapuhelimelle. Mikäli käytetään piiriyhteyttä, tulee IP-osoitteen lisäksi matkapuhelimeen syöttää myös Internet-soittosarjan tiedot: puhelinnumero, käyttäjätunnus ja salasana.

Joihinkin matkapuhelimiin on mahdollista noutaa WAP-asetukset SMS-lyhytviestinä, jolloin käyttäjän ei tarvitse kirjoittaa asetuksia puhelimeensa.

WAP-välityspalvelin osaa välittää WML-muotoisen hypertekstidokumentin lisäksi muun muassa WBMP- kuvatiedostoja (Wireless Bitmap) ja WMLScript-sovelluksia.

WAP-dokumentteja jakavana palvelimena voi olla tavallinen WWW-palvelin, kunhan sille voidaan määrittää WAP:n edellyttämät sisältötyypit. Tiedostojen sisällöt erotetaan toisistaan MIME-sisältötyypimäärittelyn mukaan (content-type), aivan kuten WWW-tekniikassakin.

| Tiedosto | Tiedostopääte | Sisältötyyppi |
|--------------------------|---------------|--------------------------------|
| WML-dokumentti | .wml | text/vnd.wap.wml |
| Wireless Bitmap | .wbmp | image/vnd.wap.wbmp |
| Käännetty WML-dokumentti | .wmlc | application/vnd.wap.wmlc |
| WMLScript | .wmls | text/vnd.wap.wmlscript |
| Käännetty WMLScript | .wmlsc | application/vnd.wap.wmlscriptc |

Taulukko 2.1 WAP-sisältötyypit (MIME)

Koska WAP-dokumentit (*.wml, *.wmls jne) eroavat muusta WWW-palvelimelta lähtevästä datasta vain tiedostopäätteeltä ja sisältötyypinmäärittelykseltä, voidaan useita olemassa olevia dynaamisia sovelluksia, muun muassa CGI-ohjelmia, käyttää hyvin pienin muutoksin myös matkapuhelimella luettavan datan tuottamiseen. Useat dynaamiset sovellukset voidaan näin ollen toteuttaa siten, että samasta tiedosta voidaan tuottaa automaattisesti kaksi eri versiota, toinen mobiilikäyttäjää ja toinen WWW-käyttöä varten.

```
#!/usr/bin/perl

print "Content-type: text/vnd.wap.wml\n\n";

print "<?xml version=\"1.0\"?>\n";
print "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\" \"\n";
print "\"http://www.wapforum.org/DTD/wml_1.1.xml\">\n";

print "<wml>\n";
print "<card id=\"korttil\" title=\"otsikko\">\n";
print "<p>Hello world!</p>\n";
print "</card>\n";
print "</wml>\n";
```

Esimerkki 2.1 WAP-dokumentin luominen PERL-ohjelmointikielellä



Kuva 2.1 Perl-ohjelmointikielellä luotu esimerkki WAP-sovelluksesta matkapuhelimen näytöllä

Kuvan näkymä on edellyttänyt matkapuhelimelta yhteyttä välityspalvelimeen, jonka kautta edelleen WWW-palvelimeen, jossa CGI-ohjelma (oheinen Perl-listaus) sijaitsee. WWW-palvelin on asetustiedostonsa mukaisesti suorittanut Perl-kielisen ohjelman, ja tulostanut tulosteen standardivirtaan, josta se on ohjautunut takaisin välityspalvelimelle ja sitä kautta matkapuhelimen WML-selaimelle.

3 Wireless Markup Language

Wireless Markup Language (WML) on erityisesti kannettavia päätelaitteita varten suunniteltu sivunkuvauskieli. WML:ää suunniteltaessa on otettu huomioon erityisesti WAP-päätelaitteen näytön pienikokoisuus, hitaina siirtonopeuksina näkyvä datatien kapea kaista sekä päätelaitteen rajalliset muisti- ja prosessorikapasiteetit. WML-kielessä on myös pyritty ottamaan huomioon syöttölaitteiden rajoitteet. WAP-päätelaitteen tekstinsyöttö saattaa hoitua matkapuhelimen näppäimistöltä tai esimerkiksi kosketusnäytön ja tätä varten suunnitellun kynän avulla.

XML-standardiin (EXtensible Markup Language) pohjautuva WML käsittää mahdollisuudet tekstiä ja kuvaa sisältävien WAP-dokumenttien esittämiseen. WML-standardi sisältää määrykset tekstin korostamiseen ja muotoiluun. Dokumentteihin on lisäksi mahdollista upottaa erilaisia ohjausrakenteita, muun muassa valintaikkunoita ja tekstinsyöttöeditoreja.

WML vaikuttaa HTML:n kevytversiolta sovitettuna matkapuhelimen tai PDA-laitteen näytölle. Sen lisäksi, että kieltä on HTML:ään verrattuna kevennetty, on siihen lisätty joitakin toimintoja, mitä HTML ei tue, esimerkkinä näistä mainittakoon muuttujamäärittelyt.

Wireless Markup Language ei WWW:ssä käytettävän HTML:n tavoin määritä sivuille tarkkaa ulkoasua, vaan lopullinen muotoilu jää käyttäjän selainohjelman päätettäväksi. WML-dokumentit ovat rakenteellisia ja pyrkivät vain toimittamaan sisältönsä mahdollisimman laajalle lukijakunnalle. Dokumenteilla ei täten pyritä tavoittelemaan yhtenäistä ulkoasua laitteistosta riippumatta. Dokumenttien yhtenäisen ulkoasun sijaan SGML-metakielen (Standard Generalised Markup Language) pohjalta muodostetulla WML-määryksellä pyritään takaamaan järjestelmästä riippumaton tapa kuvata dokumentin rakenne ja sisältö.

WML-kielen yksittäiset sivut, kortit, muodostavat laajemman kokonaisuuden, jota nimitetään pakaksi (deck). Päätelaitteesta ja sen näyttöresoluutiosta riippuen käyttäjä näkee ruudullaan yhden tai useamman kortin. Näytön koosta riippumatta käyttäjän on mahdollista liikkua pakan sisältämien korttien välillä käyttäen hyperlinkkejä. WML-pakka on URL-osoitteeltaan yhtäläinen HTML-sivun kanssa. WML-korttien käyttöä WAP-tekniikassa voisi verrata HTML:n <anchor>-tagiin, jolla voidaan määrittää siirtyminen tiettyyn kohtaan dokumentin sisällä.

WAP-sivujen välille voi HTML:n tapaan luoda hyperlinkkejä. Linkit voivat osoittaa joko johonkin korttiin toisessa pakassa tai vaihtoehtoisesti toiseen korttiin samassa pakassa. Hyperlinkkien tehokkaalla käytöllä mahdollistetaan laajojenkin sisältökokonaisuuksien julkaiseminen kannettavassa, pienitehoisessa päätelaitteessa. Käyttäjän selaimen lähetettävä tieto voidaan tarkoin rajata käyttämällä voimakkaasti hierarkkisia rakenteita.

WML-kieli tulkitsee isot ja pienet kirjaimet eri merkeiksi. Tämän vuoksi esimerkiksi muuttuja "Esimerkki" on aivan eri muuttuja kuin muuttuja "esimerkki". Kaikki komennoiksi tarkoitetut merkit tulee WML 1.1-standardin mukaan kirjoittaa pienin kirjaimin.

3.1 WML:n sisältötyypit ja merkistöt

WAP-päätelaite tunnistaa vastaanottamansa tiedoston sisällön ns. MIME-sisältötyypin (Content-type) avulla luvussa 2 esittämälläni tavalla. WWW-palvelin tutkii tiedostopäätettä ja asettaa sisältötyypin tiedostopäätteen mukaan. WML-tekstidokumentin MIME-sisältötyyppi on "text/vnd.wap.wml".

WML käyttää XML:n tapaan ISO/IEC-10646-merkistöä. Tällä hetkellä tämä merkkikoodaus on identtinen Unicode 2.0:n kanssa. Mikäli kaikkia dokumentin esittämiseen tarvittavia merkkejä ei voida tämän merkkikartan avulla esittää, voidaan käyttää &-alkuisia erikoismerkkikodeja puuttuvien merkkien esittämiseen (esimerkiksi ").

WML-standardi ei määritä päätelaitteen WAP-selaimelle mitään pakollista merkkimuunnosta (character encoding). WML-selain voi selvittää dokumentin käyttämän merkkimuunnoksen tiedoston alussa olevasta Content-type-otsikkorivistä, dokumentin sisältämästä metatiedosta, XML-määrittelyn pohjalta tai vaihtoehtoisesti se saattaa olettaa, että tekstipohjaisen tiedoston merkkimuunnos on ISO-8859-1 ja muihin tiedostoihin sovelletaan esimerkiksi UTF-8-muunnosta.

3.2 WML päätelaitteessa

WAP-päätelaitteet eivät ainakaan toistaiseksi osaa tulkita ASCII-muotoista WML-koodia sinällään, vaan koodi tulee ensin muuttaa binäärimuotoon. WAP-välityspalvelin tekee tämän muunnoksen luvussa 2 esittämälläni tavalla. Käännetyn binäärimuotoisen WML-koodin tulkitseminen vaatii päätelaitteelta vähemmän tehoa ja tiedon siirtäminen on nopeampaa.

WAP-päätelaite voi asettaa rajoituksia esittämäänsä sisältöön ja siten sivujen ulkoasuun. Kulloinkin käytettävän päätelaitteen muisti- ja prosessorikapasiteetit rajoittavat laitteella esitettävien WML-dokumenttien pakkakokoa. Esimerkiksi tämänhetkissä (07/2000) matkapuhelimissa käännetyn WML-pakan (*.wmlc jne) maksimikoko on 1300-4000 tavua.

WAP-laite määrittää sivujen ulkoasun ja selain päättää sivun lopullisen esitysasun. Jo sivun suunnitteluvaiheessa on pyrittävä muistamaan se tosiasia, ettei laitteistosta riippumatonta dokumenttia voida näyttää kaikilla laitteilla samannäköisenä. Vaikka sisältö olisi täsmälleen sama, on ulkoasussa havaittavissa suuriakin vaihteluita.

WAP-päätelaitteiden WML-selaimet pyrkivät näyttämään dokumentit juuri kyseiselle laitteelle parhaalla mahdollisella tavalla. Kaikkia toimintoja ei kuitenkaan välttämättä voida rajallisen muistitilan vuoksi toteuttaa, jolloin esimerkiksi tekstin tasaukset eivät näy selaimessa oikein, tai selain ei ymmärrä taulukkomuotoista tekstiä. Vaikka WML-standardin versio 1.1 esittelee esimerkiksi lukuisia tekstin korostuskeinoja, ei ole sanottua, että kaikki WML1.1-päätelaitteet tukisivat niitä kaikkia.

3.3 WML-kielen komennot

Aloitamme WML-kieleen tutustumisen suoraan esimerkin avulla. Oheinen listaus tuottaa WAP-selaimen ruudulle yksinkertaisen, vain tekstiä sisältävän WML-sivun. WML-sivunkuvauskielessä on havaittavissa selvä yhteys HTML-kieleen.

```
01 <?xml version="1.0"?>
02 <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
03 "http://www.wapforum.org/DTD/wml_1.1.xml">

04 <wml>
05 <card id="korttil" title="Kortin otsikko">
06     <p>
07         Kortin varsinainen teksti..
08     </p>
09 </card>
10 </wml>
```

Esimerkki 3.1 Yksinkertainen WML-listaus

Rivit 1, 2 ja 3 (rivinumerot ovat listauksen vasemmassa reunassa, rivinumeroita ei kirjoiteta varsinaiseen WML-dokumenttiin) sisältävät WML-dokumentille pakollisen otsikkotiedon (header) ja dokumenttimäärittelyn. Näillä riveillä kerrotaan XML- ja WML-standardien versionumerot sekä DTD:n sijainti URL-osoitteena. WML-dokumentin tulee aina alkaa oheisen kaltaisilla otsikkoriveillä.

Rivillä 4 aloitetaan varsinainen WML-dokumentti (eli pakka) <wml>-tagilla. <wml>-aloitustagia vastaa rivillä 10 oleva <wml>-lopetustagi </wml>, vastaavasti kuin HTML-kielessä.

Rivi 5 kertoo dokumentin sisältämän ensimmäisen kortin tiedot ja avaa kortin. Kortille voidaan antaa otsikko (title), joka oheisessa esimerkissä on *Kortin otsikko*. Kortin tunniste (id) on määrite, jolla kortti voidaan yksilöidä kunkin pakan (WML-dokumentin) sisällä. Tällöin on mahdollista käyttää esimerkiksi hyperlinkkiä osoittamaan yksittäiseen korttiin. Oheisessa listauksessa näkyvään korttiin viitattaisiin esimerkiksi osoitteella

<http://localhost/esim1.wml#kortti1> olettaen, että dokumentti on paikallinen ja nimeltään esim1.wml.

Rivillä 6 aloitetaan uusi kappale, jonka sisältöä kortin varsinainen teksti rivillä 7 on. Kappale päätetään rivillä 8 kappaleen lopetustagiin `</p>`. WML-dokumentin sisällön tulee olla jaettu kappaleisiin, eikä kappaleiden ulkopuolella saa olla sisältöä. Rivit 9 ja 10 päättävät kortin `</card>` ja dokumentin `</wml>`.



Kuva 3.1 Esimerkki WML-dokumentista matkapuhelimen WAP-selaimessa

3.3.1 WML-pakka, kortit ja WML-dokumenttien kappalejako

WML-pakan avaamiseen ja sulkemiseen käytetään `<wml></wml>` -tageja. Jokaisen WML-pakan tulee alkaa `<wml>`-tagilla ja loppua `</wml>`-tagiin.

Uuden kortin alkuun kirjoitetaan `<card>`-tagi. `<card>`-tagille voidaan antaa argumentteina korttikohtainen tunnus (id) ja kortin otsikko. Lisäksi voidaan määritellä halutaanko kortti päivittää jokaisen kyselyn yhteydessä (`newcontext=true`) vai sallitaanko käyttäjän selaimeen jääneiden muuttujien ym tiedon käyttäminen uudelleen (`newcontext=false`).

```
<card  
id="nimi"  
title="otsikkorivi"  
newcontext="true/false"  
ordered="true/false"  
onenterforward="URL"  
onenterbackward="URL"  
ontimer="URL">  
kortin sisältö  
</card>
```

Mikäli selaimelle halutaan sallia vapaus järjestellä dokumentin sisältämiä elementtejä, voidaan määre *ordered* asettaa epätodeksi, eli käyttää määritystä `ordered=false`.

Mikäli selaimen `<go>`-toiminnon avulla sivulle tuleva käyttäjä halutaan siirtää suoraan jollekin toiselle sivulle, voidaan tämä tehdä `onenterforward`-määrityksen avulla. `Onenterforward` tarvitsee parametrikseen URL-osoitteen. `Onenterbackward` toimii samankaltaisesti, mutta siinä tapauksessa, että käyttäjä käyttää selaimen takaisin-nappulaa. `Ontimer`-toiminnolla voidaan

ajastetusti siirtyä tiettyyn URL-osoitteeseen. Tällöin Ontimer-tapahtuma käynnistetään timer-tagin osoittaman ajan täytyessä. Onenter- ja ontimer-tapahtumista myöhemmin lisää.

Kappalejakoon käytetään WML-kielessä <p>-tagia. <p>-tagilla voi olla attribuutti *align*, jolla määritetään kappaleen tasaus. Mikäli määritetään align="right", tasataan kappale oikeaan reunaan, mikäli taas align="left", tasataan kappale vasempaan reunaan. Align voi saada myös arvon center, jolloin teksti keskitetään. Mode-määrittelyllä <p align="tasaus" mode="rivitys"> kappaleen sisältö </p> voidaan asettaa parametrit wrap ja nowrap, joista ensimmäinen sallii selainohjelmalle rivityksen. Nowrap-parametrin tapauksessa dokumentin tuottajan pitäisi huolehtia rivityksestä.

Mikäli WML-koodin sekaan halutaan lisätä kommentti, tapahtuu se HTML-kielen tapaan seuraavan syntaksin avulla:

```
<!-- kommentti -->
```

3.3.2 WML-pakan <head>-osio ja metatieto

WML-tiedoston head-osiossa voidaan määrittellä sellaista selainohjelmalle välittyvää tietoa, jota ei kuitenkaan näytetä päätelaitteen ruudulla. Head-osiossa voidaan myös rajata WML-pakan selailua <access>-elementillä.

Head-osio aloitetaan tagilla <head> ja päätetään vastaavasti tagiin </head>. Näiden väliin tulee määrittää ainakin yksi elementti, joko <access> tai <meta>. <access>-elementillä voidaan rajata pääsyä kyseiselle WML-pakalle. <access> ottaa määreinään domain-osoitteen (domain) ja polun (path). Domain ilmoittaa <access>-elementille sen osoitepääteen eli domain-osoitteen, josta tuleva käyttäjä päästetään kyseiselle pakalle. Path-argumentti kertoo sen polun, missä sijaitsevalta sivulta käyttäjän pääsy sallitaan.

Oheiset esimerkit havainnollistanevat komentojen käyttöä.

```
<head>
<access domain="minunosoite.com" path="/WAP-esimerkit/">
</head>
```

Esimerkki 3.2 <access>-elementin käyttö

Esimerkin tapauksessa käyttäjä voi tulla lukemaan pakan sisältämiä kortteja osoitteessa minunosoite.com sijaitsevan palvelimen alihakemistosta *WAP-esimerkit* tai sen alihakemistoista löytyviltä WML-sivuilta. Esimerkiksi osoite <http://www.minunosoite.com/WAP->

esimerkit/wapdemo.wml kelpaa, mutta osoitteet <http://www.jonkunmuun.com/wap.wml> tai <http://www.minunosoite.com/WAPPI/> eivät kelpaa.

<meta>-tagilla voidaan määrittää selaimelle esimerkiksi pakan kirjoittamiseen käytetty merkitö, kuten seuraavassa esimerkissä tehdään.

```
<head>
<meta content="charset" user agent="character-set=UTF-8"/>
</head>
```

Esimerkki 3.3 <meta>-tagin käyttö

<meta>-tagin saamat argumentit vaihtelevat *content*-määreen mukaan. Meta-tietoa voidaan hyödyntää selaimen lisäksi myös muun muassa hakuagenteissa.

3.3.3 Tekstin muotoilu, korostus ja erikoismerkkien käyttö

WAP-dokumenttiin voidaan luoda rivinvaihto käyttäen
-tagia. Koska
 on ns. tyhjä tagi, eli sille ei ole lopetustagia, tulee tagiin sisällyttää kauttaviiva.

WML-kieli sallii muutamia keinoja tekstin korostamiseen. WML-selaimet (matkapuhelinten WAP-selaimet) tukevat seuraavia korostuskeinoja vaihtelevasti.

Seuraavassa WML-kielen korostuskeinoja.

| | |
|---------------------------|---|
| teksti | Korostettu teksti |
| teksti | "Voimakas" teksti, näytetään usein lihavointina |
| teksti | Lihavointi |
| <i> teksti </i> | <i>Kursivointi</i> |
| <u> teksti </u> | <u>Alleviivaus</u> |
| | |
| <small> teksti </small> | Pienempi kirjasinkoko |
| <big> teksti </big> | Suurempi kirjasinkoko |

WML sisältää HTML-kielen tavoin erityiset &-alkuiset koodit erikoismerkeille. Seuraavassa lyhyt listaus tärkeimmistä erikoismerkkikoodeista.

Erikoismerkkejä

| | |
|--------|-------------------------------------|
| " | " |
| & | & |
| ' | ' |
| < | < |
| > | > |
| | välilyönti (non-breaking space) |
| ­ | ohjeellinen tavuviiva, tavutusvihje |

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="korttil" title="korostuksia">
    <p align="center">

      <b>lihavoitu</b><br/>
      <i>kursivoitu</i><br/>
      <u>alleviivattu</u><br/>

    </p>
  </card>
</wml>
```

Esimerkki 3.4 WML-kielen korostustagien käyttö



Kuva 3.2 Esimerkiksi Nokia 7110:n WAP-selain ei osaa näyttää korostustageja.

3.3.4 Template

Template-rakenteella on mahdollista tehdä koko pakkaa koskevia tyylimäärittäyksiä. <template>-määrittäyksen syntaksi on seuraava:

```
<template onenterforward="URL" onenterbackward="URL" ontimer="URL">
sisältö </template>
```

Esimerkki <template>-toiminnon käytöstä myöhemmin tehtävien ja tapahtumien yhteydessä.

3.3.5 WML ja hyperlinkit

WML-kielinen hyperlinkki voidaan toteuttaa kahdella tavalla. Hyperlinkin luomiseen voidaan käyttää HTML-kielen kaltaista `<a href...>`-rakennetta tai vaihtoehtoisesti hyperlinkki voidaan luoda `<anchor>`-elementin avulla. Anchor on toimintona laajempi ja vaatii parikseen elementin, jolla määritetään linkin aktivoituessa suoritettava toiminto.

`<anchor>`-muotoinen linkki koostuu kahdesta osasta, `<anchor>`-rakenteesta ja sen suorittamasta tehtävästä. `<anchor>`-elementin edellyttämä rakenne käy ilmi seuraavasta esimerkistä.

Tehtävänä voi olla joko refresh, go tai prev.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="kortti1" title="Linkkiesimerkki">

<p>
<anchor title="Linkki">Linkki
<go href="toinensivu.wml">
</go>
</anchor>
tekstin seassa.
</p>
</card>

</wml>
```

Esimerkki 3.5 WML-kielisen hyperlinkin toteuttaminen `<anchor>`-rakenteella

``-muotoinen linkki on täysin samannäköinen kuin HTML-kielessä. Linkki aloitetaan `<a>`-tagilla, jolle annetaan määreenä `href="URL"`. URL on `<a>`-tagin ainoa pakollinen määre. Linkkinä oleva teksti tulee `<a>`-aloitustagin ja ``-lopetustagin väliin.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="kortti1" title="Linkkiesimerkki">

<p>
<a href="toinensivu.wml">Linkki</a> tekstin seassa.
</p>
</card>

</wml>
```

Esimerkki 3.6 WML-kielisen hyperlinkin toteuttaminen `<a href...>`-rakenteella



Kuva 3.3 Molemmat edellä esitetyt hyperlinkkiesimerkit näyttävät selaimessa tältä.

Linkillä voidaan viitata absoluuttiseen osoitteeseen, esimerkiksi `http://wap.firma.fi/index.wml`, tai vaihtoehtoisesti suhteelliseen osoitteeseen, esimerkiksi samassa hakemistossa olevaan dokumenttiin `dokumentti2.wml`. Linkki voi osoittaa myös toiseen korttiin samassa pakassa.

Mikäli linkin halutaan osoittavan toiseen korttiin samassa pakassa, on osoite muotoa `dokumentti.wml#kortti`. Vaihtoehtoisesti voidaan osoitteena käyttää pelkkää kortin tunnusta, jolloin kortin tunnistemerkki `#` vaaditaan tunnuksen edessä.

Esimerkkejä:

<a>-tagi:

```
<a href="#kortti1">Linkki korttiin 1</a>

<a href="http://wap.org/dokumentti.wml#kortti1">
Kortti1 muualla</a>

<a href="http://wap.org/dokumentti.wml">Linkki pakkaan
dokumentti.wml</a>
```

<anchor>-tagi:

```
<anchor title="Linkki">Linkki korttiin 1
<go href="#kortti1">
</go>
</anchor>

<anchor title="Linkki">Kortti 1 muualla
<go href="http://wap.org/dokumentti.wml#kortti1">
</go>
</anchor>

<anchor title="Linkki">
Linkki pakkaan dokumentti.wml
<go href="http://wap.org/dokumentti.wml">
</go>
</anchor>
```

3.3.6 Kuvat WML-dokumentissa

WML-dokumenteissa voidaan käyttää WBMP-muotoisia (Wireless Bitmap) kuvatiedostoja. Kuva lisätään WML-dokumenttiin -tagilla. Yksinkertaisimmillaan kuva saadaan näkyviin seuraavanlaisella rivillä:

```

```

Mikäli WAP-selain ei voi näyttää kuvaa (kuvia ei tueta, tai kuva on liian iso), näyttää selain alt-määrittelyksellä sisällytetyn tekstin. Alt-teksti on WML-standardin mukaan pakollinen määre.

-tagille voidaan antaa myös muita määreitä. Seuraavassa kaikki -tagin määreet.

| | | |
|-----------|-----------------------|--|
| src= | "URL" | URL-osoite |
| alt= | "alt-teksti" | kuvalle vaihtoehtoinen teksti |
| localsrc= | "ikoni" | WAP-laitteen (paikallinen) ikoni |
| align= | "sijoittelu" | kuvan sijoittaminen ruudulla |
| | | <ul style="list-style-type: none">• left = vasen reuna• right = oikea reuna• center = keskitetty |
| vspace= | "tila_pystysuunnassa" | kuvalle varattava tila pikseleinä |
| hspace= | "tila_vaakasuunnassa" | kuvalle varattava tila pikseleinä |
| height= | "korkeus" | kuvan korkeus |
| width= | "leveys" | kuvan leveys |

```
<?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"  
"http://www.wapforum.org/DTD/wml_1.1.xml">  
  
<wml>  
  <card id="card1" title="Kuvaesimerkki">  
    <p>  
        
    </p>  
  </card>  
</wml>
```

Esimerkki 3.7 WML-lähdekoodi kuvan *pallo.wbmp* näyttämiseksi



Kuva 3.4 Esimerkki WBMP-kuvan näyttämisestä (WML-lähdekoodi yllä). Kuvassa oleva WAP-selain keskittää kuvan automaattisesti.

3.3.7 Taulukot

```
<table  
  
title="otsikko"  
align="tasaus"  
columns="lukumäärä">  
  
<tr>  
<td>solun teksti </td>  
</tr>  
  
</table>
```


Taulukko luodaan <table>-tagilla. Taulukolle voidaan määrittää otsikko *title*-määreellä. Taulukon tasaus voidaan asettaa *align*-attribuutilla (left,right,center), kuten <p>-tagin yhteydessä. *Columns*-määre kertoo taulukolle käytettävien sarakkeiden määrän.

Uuden rivin luominen onnistuu <tr>-tagilla ja uusi sarake aloitetaan <td>-tagilla.

Seuraavassa esimerkki taulukon käytöstä.

```
<?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"  
"http://www.wapforum.org/DTD/wml_1.1.xml">  
  
<wml>  
  
<card id="kortti1" title="Esimerkki taulukosta">  
  
<p>  
  <table columns="2" align="left">  
  
    <tr>  
      <td>Objekti </td><td>Kuva </td>  
    </tr>  
  
    <tr>  
      <td>Pallo</td><td></td>  
    </tr>  
  
  </table>  
  
</p>  
</card>  
</wml>
```

Esimerkki 3.8 Taulukon luominen WML-kielellä

| Objekti | Kuva |
|---------|---|
| Pallo |  |

Taulukko 3.1 WML-selaimen muotoilema taulukko

WML-kielinen taulukko voi sisältää tekstiä ja/tai kuvaa, mutta ei toista (sisäkkäistä) taulukkoa.

3.3.8 Lomakkeet

Lomakkeet ovat interaktiivisten WAP-sovellusten peruselementtejä. Lomakkeen avulla sovelluksen käyttäjältä voidaan kysyä esimerkiksi käyttäjätunnus ja salasana, tai lomaketta voidaan käyttää gallup-kyselyn toteuttamiseen valintalistojen avulla. Lomakkeen sisältö voidaan ohjata joko WMLScript-kielen käsiteltäväksi tai lomakkeen tietoja voidaan käyttää CGI-rajapinnan avulla.

Käsittelen tässä lyhyesti lomakkeiden ohjelmoinnin WML-kielellä syventymättä kuitenkaan tarkemmin lomakkeiden käsittelyyn. WWW-tekniikan kaltaisesti WAP-lomakkeiden tieto siirretään sitä käsittelevälle ohjelmalle joko POST- tai GET-metodilla.

```
<input
name="muuttuja"
title="otsikko"
type="tyyppi"
value="oletusarvo"
format="muoto"
emptyok="true tai false"
size="koko"
maxlength="maksimipituus"
tabindex="n"/>
```

`<input>`-tagin avulla voidaan luoda tekstinsyöttöikkuna, jonka tyyppi (type) on joko teksti (text) tai salasana (password). Mikäli kentälle valitaan tyypiksi salasana, näyttää selain käyttäjän kenttään syöttämän tekstin esimerkiksi asteriskeina (*). Muutoin syötetty teksti näkyy näytöllä. Kentälle tulee antaa nimi, jonka mukaan nimetään myös kentän sisältömuuttuja. Kenttä nimetään name-attribuutilla.

Mikäli kenttään halutaan sijoittaa jokin oletusarvoinen merkkijono, tulee se asettaa argumentilla value. Size ja maxlength-optioilla voidaan määritellä kentän koko ja merkkijonon maksimipituus. Mikäli tyhjän kentän lähettäminen halutaan estää, voidaan antaa argumentti emptyok="false". Tabindex-argumentin avulla dokumentin sisältämät tekstikentät voidaan järjestää siten, että käyttäjän selatessa dokumenttia alaspäin, tulee hän liikkuneeksi kentät tietyssä järjestyksessä.

`<input>`-tagin avulla voidaan luoda esimerkiksi seuraavan esimerkin kaltainen tekstinsyöttöeditori.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="kortti1" title="Lomake">

<p>
Tekstikenttä:
<input type="text" name="teksti"/>
</p>
</card>
</wml>
```

Esimerkki 3.9
Input-tagin ja
tekstikenttä



Kuva 3.5 Tekstikenttä WAP-selaimessa

<input>-kentän syötteen muotoa voi ohjata määrittämällä input-tagille format-argumentin sopivin määrein. Seuraavassa taulukko muutamista muotoiluvaihtoehdoista.

| Format-kentän argumentti, esimerkiksi | Tarkoitus |
|---------------------------------------|--|
| <input ... format="NN"> | |
| NNNNNN | Syötteen tulee olla kuusi merkkiä pitkä numero. |
| MMM | Syötteen tulee olla kolmikirjaiminen merkkijono. |
| *N | Syötteen tulee olla numero. |
| *M | Syöte voi olla mikä tahansa merkkijono. |
| XXX | Syötteen tulee olla kolme isoa kirjainta. |
| *X | Syötteen tulee olla suuraakkosin. |
| *x (pieni x) | Syötteen tulee olla pienaakkosin. |

Taulukko 3.2 Format-argumentin määreitä

Tekstisyöttökentän lisäksi WML-kielellä on mahdollista luoda valintalistoja. Valintalistan luominen onnistuu <select> ja <option>--komennnoilla.

Valintalista sijoitetaan <select></select>-tagien väliin. Aloitustagi <select> voi saada seuraavia määreitä:

| | |
|---------------------------|--|
| title="otsikko" | valintalistan otsikko |
| multiple="true tai false" | voidaanko valita useita valintoja (tosi/epätosi) |
| name="muuttuja" | valintalistan nimi (muuttujan nimi) |
| value="oletus" | oletuksena valittu valinta |
| iname="i_muuttuja" | indeksimuuttuja, sisältää valinnat numeroina |
| ivalue="oletus" | oletuksena valitun valinnan numero (indeksi) |
| tabindex="numero"> | elementin järjestysnumero selatessa |

<option>-tagilla valintalistaan lisättävän valinnan muoto on seuraava:

<option title="otsikko" value="muuttuja" onpick="URL"> valinnan_nimi </option>

Mikäli onpick-määrettä käytetään ja sille annetaan argumenttina URL-osoite, siirtyy selain kyseiseen URL-osoitteeseen valinnan tapahduttua. Onpick käsitellään tapahtumien yhteydessä.

Seuraavassa esimerkki valintalistan käytöstä.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="card1" title="Valintalista">

<do type="accept" label="OK">
<go method="post" href="http://localhost/cgi-bin/handle.pl">
<postfield name="valinta" value="$(valinta)"/>
</go>
</do>

<p>
Tee valinta:
<select name="valinta">
<option value="balls">palloja</option>
<option value="cubes">kuutioita</option>
<option value="toroids">toroideja</option>
</select>
</p>
</card>
</wml>
```

Esimerkki 3.10 Valintalistan luominen WML-kielellä



Kuva 3.6 Valintalistan käyttö WAP-selaimessa

Oheisen esimerkin tapauksessa käyttäjän tekemä valinta välitetään muuttujan \$valinta avulla paikallisen Perl-ohjelman (handle.pl) käsiteltäväksi.

<select>-valintalistan valintoja voidaan tarvittaessa ryhmitellä <optgroup>- ja <fieldset>-tagien avulla. <optgroup>-tagi sallii valintojen järjestämisen ryhmiksi ja <fieldset>-tagilla voidaan muodostaa kokonaisuuksia, joihin voi kuulua valintalistaelementtien lisäksi myös muita osia.

<optgroup>-tagia voidaan käyttää seuraavasti:

```
<select name="object">
  <optgroup title="2D">
    <option value="circle">circle</option>
    <option value="line">line</option>
  </optgroup>

  <optgroup title="3D">
    <option value="torus">torus</option>
    <option value="sphere">sphere</option>
    <option value="cylinder">cylinder</option>
  </optgroup>
</select>
```

Esimerkki 3.11
<optgroup>-tagin
käyttö

Vastaavanlainen jako voidaan suorittaa myös <fieldset>-tagilla, mutta <fieldset> sallii <option>-tagin lisäksi myös muita elementtejä:

```
<fieldset title="2D">
  Haluamasi objekti:
  <input type="text" name="obj" format="*M"/><br/>

  <select name="kaksiul">
    <option value="circle">circle</option>
    <option value="line">line</option>
  </select>
</fieldset>

<fieldset title="3D">
  Haluamasi objekti:
  <input type="text" name="obj" format="*M"/><br/>
</fieldset>
```

Esimerkki 3.12
<fieldset>-tagin
käyttö

3.3.9 Tapahtumat (events)

Tapahtumat tarkoittavat toimintoja, jotka aktivoituvat tietyn tapahtuman yhteydessä, esimerkiksi kortille saavuttaessa tai kortilta poistuttaessa. Tällaisiksi tapahtumiksi lasketaan myös timer-ajastimen täytyminen ja valinnan tekeminen valintaikkunassa.

do

do-toiminnon avulla voidaan liittää toiminto selaimen määrittämään näppäimeen. Selain voi sijoittaa toiminnon esimerkiksi toimintonäppäimen valintalistaan tai sille voidaan varata oma näppäimensä. Seuraavassa esimerkissä luodaan linkki pakkaan sivu2.wml. go-tagista tarkemmin seuraavassa luvussa.

```
<do type="accept" label="Seuraava sivu">
  <go href="sivu2.wml"/>
</do>
```

Esimerkki 3.13 Do-tapahtuman käyttö <go>-tagin yhteydessä

do-toiminto voi liittyä joko pakkaan, jolloin se sijoitetaan <template>-määrittelyn sisään, tai vaihtoehtoisesti korttiin, jolloin se tulee olla <card></card>-komentojen välissä.

ontimer

ontimer-tapahtuma mahdollistaa ajastimella aktivoitavan tapahtuman käytön. Ajastimen käytöstä lisää ajastinta käsittelevässä luvussa 3.3.11.

onenterforward ja onenterbackward

onenterforward- ja onenterbackward-tapahtumat mahdollistavat käyttäjän ohjaamisen toiselle sivulle, mikäli tämä on käyttänyt kortille tulemiseen selaimen eteenpäin- (forward) tai taaksepäin- (backward) näppäintä.

Onenterbackward- tai onenterforward-tapahtuma voidaan sijoittaa joko <card>-määrittelyn tai <template>-määrittelyn sisälle. Käytettäessä korttikohtaista <card>-määrittystä, kohdistuu tapahtuma vain tietylle kortille, <template> sen sijaan vaikuttaa oletuksena pakan kaikkiin kortteihin.

Seuraavassa esimerkissä eteenpäin-toiminnolla saapuva käyttäjä ohjataan sivulle toinen.wml.

```
<card id="kortti1" title="Hienoa!" onenterforward="toinen.wml">
  <p>Et käyttänyt selaimesi eteenpäin-toimintoa.</p>
</card>
```

Esimerkki 3.14 Onenterforward-tapahtuman käyttö

onpick

onpick-tapahtuma sijoitetaan <option>-tagin attribuutiksi, jolloin selain siirretään määriteltyyn URL-osoitteeseen käyttäjän valitessa kyseinen valinta. Seuraavassa esimerkki onpick-tapahtumasta.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="korttil" title="onpick-esimerkki">
<p>
  Tee valinta:
  <select name="dokumentti">
<option value="1" onpick="pallot.wml">Tutki palloja</option>
<option value="2" onpick="kuutiot.wml">Tutki kuutioita</option>
</select>
</p>
</card>
</wml>
```

Esimerkki 3.15 Onpick-toiminnon käyttö valintalistan yhteydessä

onevent

Onevent sallii tapahtuman liittämisen useisiin erilaisiin elementteihin. Sitä voidaan käyttää esimerkiksi <go>-tehtävän yhteydessä. Onevent-rakenne on seuraava:

```
<onevent type="tyyppi"> tehtävä </onevent>
```

postfield

Postfield-toiminnolla määritellään muuttuja ja muuttujan arvo, jotka myöhemmin voidaan lähettää palvelimelle. Postfield-tapahtumaa käytettiin lomakkeiden yhteydessä tiedon välittämiseen. Postfield-tapahtuman syntaksi on seuraava:

```
<postfield name="nimi" value="arvo"/>
```

3.3.10 Tehtävät (tasks)

Tehtävillä tarkoitetaan toimintoja, jotka suoritetaan tapahtumien yhteydessä. Näitä toimintoja ovat go, prev, noop ja refresh.

go

go-toiminnolla määritellään siirtyminen toiselle kortille tapahtuman seurauksena.

```
<card id="kortti1" title="go-esimerkki">
<do type="accept" label="Seuraava">
<go href="seuraavasivu.wml"/>
</do>
</card>
```

Esimerkki 3.16 go-toiminnon käyttö

prev

<prev>-tagilla käyttäjälle voidaan antaa mahdollisuus palata edelliselle sivulle. Seuraavassa esimerkki komennon käytöstä. Esimerkissä luodaan malli, jota käytetään pakan jokaisessa kortissa. Malli määrittelee <prev/>-toiminnon näppäimeen, jolle selain antaa nimen "Edellinen sivu".

```
<template>
<do type="prev" name="back" label="Edellinen sivu">
<prev/>
</do>
</template>
```

Esimerkki 3.17 Prev-elementin käyttö

noop

noop on komento, jolla voidaan kieltää tapahtuman toiminto, esimerkiksi ylikirjoittaen olemassaoleva määrittely. noop-komento on tarpeellinen, kun pakan kaikkiin kortteihin tehty <template>-määrittely halutaan tietyille kortille kumota.

Seuraavassa esimerkissä noop-komentoa käytetään olemassaolevan määityksen kumoamiseen.

```
...
<template>
<do type="prev" name="edellinen" label="edellinen">
<prev/>
</do>
</template>

<card id="korttil">
<p>
Hei!</p>

<do type="accept" label="seuraava">
  <go href="uusi.wml"/>
</do>

<do type="prev" name="edellinen">
  <noop/>
</do>
...
</card>
```

Esimerkki 3.18 noop-toiminnon käyttö

refresh

Refresh-tagilla voidaan päivittää muuttujien arvot. Oheisessa esimerkissä alun perin oletusarvona ollut "Pallo" tyhjennetään.

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="korttil" title="refresh-esimerkki">
<p>
Objekti:
<input title="Objekti" name="objekti" value="Pallo"/>
  <do type="accept" label="refresh">
    <refresh>
      <setvar name="nimi" value=""/>
    </refresh>
  </do>
</p>
</card>
</wml>
```

Esimerkki 3.19 Refresh-tehtävän käyttö

3.3.11 Ajastin

<timer/>-elementti mahdollistaa ajastettujen toimintojen liittämisen WML-dokumentteihin. Kortille saavuttaessa ajastin käynnistetään ja se saa arvon nolla. Kortilta poistuttaessa ajastin pysähtyy. Mikäli käyttäjä ei poistu kortilta ennen kuin ajastin saavuttaa <timer/>-elementillä asetetun arvon, suoritetaan ontimer-tapahtumalla lisätty toiminto.

<timer/>-elementille annetaan parametrina ajastimen muuttujan nimi sekä arvo kymmenesosasekunteinä. Tämän arvon täytyttyä (kortille saapumisesta kulunut arvo sekuntia) suoritetaan ontimer-tapahtuma.

Ajastinta voidaan hyödyntää usealla eri tavalla, ja saman asian toteuttamiseenkin voidaan käyttää useita tapoja. Seuraavassa kaksi esimerkkiä ajastimesta.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="korttil" title="Timer-esimerkki">

<onevent type="ontimer">
  <go href="uusisivu.wml"/>
</onevent>

<timer value="50"/>
<p>
Odota hetki, niin sinut siirryt sivulle
uusisivu.wml..
</p>
</card>
</wml>
```

Esimerkki 3.20
Ajastimen käyttö
onevent-
tapahtuman
yhteydessä

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="korttil" title="Timer-esimerkki"
ontimer="uusisivu.wml">

<timer value="50"/>
<p>
Odota hetki, niin sinut siirryt sivulle
uusisivu.wml..
</p>
</card>
</wml>
```

Esimerkki 3.21
Ajastimen käyttö
<card>-elementin
yhteydessä



Kuva 3.7 Molemmat edellä esitetyt listaukset tuottavat täsmälleen samanlaisen tuloksen

3.3.12 Muuttujat

WML-kieli sisältää HTML-kielestä poiketen mahdollisuuden luoda muuttujia ja sijoittaa niihin arvoja. Muuttujien tehtävänä on välittää tietoa dokumenttien eri elementtien välillä sekä siirtää tietoa dokumentista toiseen. Lomakkeelle annettu tieto tallennetaan muuttujiin, joista tieto voidaan edelleen välittää sitä käsittelevän sovelluksen käsiteltäväksi. Muuttuja voi WML-kielessä saada arvonsa myös <setvar/>-tagin avulla. <setvar/>-tagilla on kaksi pakollista määrettä, joista ensimmäinen (name) kertoo muuttujan nimen ja toinen (value) antaa muuttujalle arvon.

Seuraavassa esimerkissä tallennetaan muuttujan *muuttuja* arvoksi *arvo*. Käyttäjän valitessa linkin, välitetään tämä muuttuja <go>-toiminnolla dokumentille "toinensivu.wml".

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="korttil" title="Tietoa muuttujaan">
<p>
<anchor title="Linkki">Linkki
  <go href="toinensivu.wml">
    <setvar name="muuttuja" value="arvo"/>
  </go>
</anchor>
</p>
</card>
</wml>
```

Esimerkki 3.22 Tiedon siirtäminen toiseen dokumenttiin muuttujan avulla

Tavallisimmat WML-käskyt

XML-otsikkotiedot ja DTD

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

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
```

WML-dokumentin ja -kortin aloitus, lopetus sekä kappalejako

| | |
|--|--|
| <code><wml></wml></code> | <code><wml></wml></code> |
| <code><head></head></code> | <code><head> sisältö </head></code> |
| <code><access></access></code> | <code><access domain="domain" path="polku" /></code> |
| <code><meta/></code> | <code><meta content="arvo" scheme="formaatti"/></code> |
| <code><card></card></code> | <code><card id="nimi" title="otsikkorivi" newcontext="true/false" ordered="true/false" onenterforward="URL" onenterbackward="URL" ontimer="URL"> .. </card></code> |
| <code><p></p></code> | <code><p align="tasaus" mode="rivitys"> kappaleen sisältö </p></code> |
| <code><template></template></code> | <code><template onenterforward="URL" onenterbackward="URL" ontimer="URL"> sisältö </template></code> |

Tapahtumat

| | |
|--|--|
| <code><do></do></code> | <code><do type="tyyppi" label="nimi" name="nimi" optional="true tai false"> tehtävä </do></code> |
| <code><onevent></onevent></code> | <code><onevent type="tyyppi"> tehtävä </onevent></code> |

```
<postfield>
value="arvo"/>
```

```
<postfield name="nimi"
```

Tekstin korostus ja erikoismerkit

**
**

Rivinvaihto

**** teksti ****

Korostettu teksti

**** teksti ****

"Voimakas" teksti, näytetään usein lihavointina

**** teksti ****

Lihavointi

<i> teksti **</i>**

Kursivointi

<u> teksti **</u>**

Alleviivaus

<small> teksti **</small>**

Pienempi kirjasinkoko

<big> teksti **</big>**

Suurempi kirjasinkoko

Erikoismerkkejä

"

"

&

&

'

'

<

<

>

>

välilyönti (non-breaking space)

­

ohjeellinen tavuviiva, tavutusvihje

Kuvat

```
<img/>
```

```

```

Tehtävät

```
<go></go>
```

```
<go href="URL-osoite"
sendreferer="true tai false"
accept-charset="merkistö"
method="metodi">
sisältö
</go>
```

```
<prev></prev>
```

```
<prev> sisältö </prev>
```

```
<noop>
```

```
<noop>
```

```
<refresh> </refresh>
```

```
<refresh> sisältö </refresh>
```

Lomakkeet

`<input/>`

```
<input name="variable"
title="otsikko" type="tyyppi"
value="arvo"
value="oletusarvo" format="mask"
emptyok="true tai false"
size="koko"
maxlength="maksimipituus"
tabindex="n"/>
```

`<select></select>`

```
<select title="label"
multiple="true tai false"
name="muuttuja" value="oletusarvo"
iname="index_var" ivalue="default"
tabindex="n">
sisältö
</select>
```

`<option/>`

```
<option title="otsikko"
value="arvo" onpick="URL">
sisältö
</option>
```

`<optgroup/>`
`title="otsikko">`

<optgroup

```
sisältö
</optgroup>
```

`<fieldset/>`
`title="otsikko">`

<fieldset

```
sisältö
</fieldset>
```

Ajastimet

`<timer/>`

```
<timer name="muuttujan_nimi"
value="arvo"/>
```

Muuttujat

`<setvar/>`

```
<setvar name="nimi" value="arvo"/>
```

Hyperlinkit

`<anchor></anchor>`

```
<anchor title="otsikko">
tehtävä (task)
text
</anchor>
```

`<a>`

```
<a title="otsikko" href="URL">
teksti</a>
```

Taulukot

`<table/>`

Uusi taulukko

`<td> </td>`

Uusi solu riville

`<tr> </tr>`

Uusi rivi taulukkoon

```
<table title="otsikko" align="tasaus" columns="sarakkeiden_lukumäärä"/>
<tr> </tr>
<td>solun teksti </td>
```


Lyhennesanasto

| | |
|-------------|---|
| CGI | Common Gateway Interface |
| DTD | Document Type Definition |
| FTP | File Transfer Protocol |
| GSM | Generic System for Mobile Telecommunication |
| GUI | Graphical User Interface |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| IP | Internet Protocol |
| RFC | Request For Comments |
| SGML | Standard Generalised Markup Language |
| SMS | Short Message Service |
| SMTP | Simple Mail Transfer Protocol |
| TCP | Transmission Control Protocol |
| UI | User Interface |
| URL | Uniform Resource Locator |
| PDA | Personal Digital Assistant |
| VoIP | Voice Over IP |
| WAE | Wireless Application Environment |
| WAP | Wireless Application Protocol |
| WBMP | Wireless Bitmap |
| WDP | Wireless Datagram Protocol |
| WML | Wireless Markup Language |
| WMLS | Wireless Markup Language Script |
| WSP | Wireless Session Protocol |
| WTLS | Wireless Transport Security Layer |
| WTP | Wireless Transmission Protocol |
| WWW | World Wide Web |
| XML | Extensible Markup Language |