

eMo
eAssari Moodle-ympäristössä
Ylläpitodokumentti

Helsinki 15.12.06
Ohjelmistotuotantoprojekti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteenlaitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9+1op/6ov)

Projektiryhmä

Järviniitty Juho

Karkulahti Ossi

Katainen Riku

Tverin Teemu

Vainio Ville-Pekka

Asiakas

Laine Harri

Vastuhenkilö

Taina Juha

Ohjaaja

Halko Antti

Kotisivu

<http://www.cs.helsinki.fi/group/emo/>

Versiohistoria

<i>Versio</i>	<i>Päiväys</i>	<i>Tehdyt muutokset</i>
0.1	06.12.2006	Pohja
0.9	13.12.2006	Sisältö
1.0	15.12.2006	Viimeistely

Sisällys

1 Johdanto.....	1
2 Asennus.....	1
3 Käyttö.....	2
4 Jäädymyksen jälkeen muuttuneet vaatimukset.....	2
5 Jäädymyksen jälkeen muuttuneet suunnitelman osat.....	2
5.1 Arkkitehtuurisuunnitelma.....	2
5.2 Komponentit.....	2
5.2.1 eMon ylliluokka.....	3
5.2.2 Tehtävätyypin lisääjä.....	3
5.2.3 Tehtävätyypin poistaja.....	3
5.2.4 Käyttöliittymä.....	3
5.2.5 Tehtävätyyppien tiedostokuvaus.....	3
5.2.6 Lokalisointi.....	4
5.2.7 Kieliriippumattomuus.....	4
5.2.8 SQL-luontilauseet.....	4
6 Toteutumattomat vaatimukset ja suunnitelman osat.....	6
7 Koodin ylläpitoon liittyvät seikat.....	6
8 Muut ylläpitoon vaikuttavat seikat.....	7

1 Johdanto

Tämä on ohjelmistotuotantoprojekti eMon tuotteen ylläpidodokumentti. Dokumentti on tarkoitettu järjestelmän jatkokehittäjille sekä ylläpitäjille avuksi järjestelmän ymmärtämisessä. Dokumentissa käsitellään järjestelmän ohjelmakoodia, sen ylläpitoa ja laajentamista ja se sisältää myös ideoita järjestelmän kehittämiseksi tulevaisuudessa. Dokumentissa otetaan myös kantaa toteutumattomiin vaatimuksiin. Tätä dokumenttia ei ole tarkoitettu luettavaksi yksinään, vaan yhdessä suunnitteludokumentin sekä järjestelmän ohjelmakoodin kanssa.

2 Asennus

eMo-järjestelmä asennetaan yksinkertaisesti siirtämällä eMo-paketin tiedostot Moodle-palvelimelle oikeisiin hakemistoihinsa. eMo toimitetaan pakkauksessa, joka osaa automaattisesti purkaa itsensä oikeisiin hakemistoihin purkuohjelmaa käytettäessä.

eMo-järjestelmän asennus Moodleen

Vaihe 1.

eMo-järjestelmän asennus vaatii pääsyn Moodlen hakemistoihin ja emo.zip -paketin, josta tarvittavat tiedostot löytyvät. Emo.zip löytyy asennus-cd:ltä tai eMo-ryhmän www-sivuilta.

Vaihe 2.

Pura emo.zip-paketti hakemistoon moodle/question/type/. Tällöin syntyy emo-alihakemisto, joka sisältää kaikki eMon käyttöön tarvittavat tiedostot. Varmista, että kaikilla eMon tiedostoilla on lukuoikeus kaikille käyttäjille, hakemistoilla pääsyoikeus kaikille käyttäjille ja kirjoitusoikeus kaikilla asentavalle käyttäjille.

Linuxissa tämän voi tehdä esim. antamalla komennon `chmod -R a+rX emo/`

Seuraavaksi kopio englannin kielen kielitiedostot hakemistosta moodle/question/type/emo/lang/en_utf8 hakemistoon moodle/lang/en_utf8

Vaihe 3.

Jos haluat käyttää Moodlea ja eMoa suomeksi, asenna ensin Moodlen ylläpitokäyttöliittymästä suomen kieli Moodleen. Tämän voi tehdä myös asennuksen aikana.

Sen jälkeen kopioi hakemistosta moodle/question/type/emo/lang/fi_utf8 kaikki tiedostot hakemistoon moodledata/lang/fi_utf8.

Vaihe 4.

Mene admin-käyttöliittymän kohtaan Ilmoitukset (Notifications), jolloin Moodle asentaa eMon ja pystyttää tietokantataulut.

Vaihe 5.

Lisätään Laatiija-rooli, jonka voi antaa henkilöille, jotka tekevät ja poistavat eMo-tehtävätyyppejä. Tämä tapahtuu hallintakäyttöliittymästä valitsemalla Käyttäjät -> Roolit -> Määritä roolit -> Lisää rooli. Anna roolille jokin nimi ja tunniste. Anna kaikkien muiden oikeuksien olla kohdassa "Peri", paitsi oikeus mod/emo:create, jonka arvoksi annetaan Salli. Lopuksi siirrä roolia ylös niin, että se on Ylläpitäjät-roolin alapuolella. Tällöin uusi rooli perii kaikki alempien roolien ominaisuudet automaattisesti.

3 Käyttö

Käyttöohjeet ovat luettavissa eMon käyttöohjeesta

(<http://www.cs.helsinki.fi/group/emo/opas/etusivu.html>)

4 Jäädetyksen jälkeen muuttuneet vaatimukset

Järjestelmälle asetetut vaatimukset eivät ole muuttuneet vaatimusmäärittelyn jälkeen. Vaatimukset on tarkistettavissa vaatimusmäärittelydokumentista ja niiden toteutuminen testausdokumentista.

5 Jäädetyksen jälkeen muuttuneet suunnitelman osat

5.1 *Arkkitehtuurisuunnitelma*

Suunnitteludokumentissa esitelty eMon oma ylikuokka on jätetty pois ja eMon kysymystyyppit kommunikoivat suoraan Moodlen liitännäisrajapinnan kanssa.

5.2 *Komponentit*

Tässä kerrotaan komponenteista, joihin on tullut muutoksia verrattuna suunnitteludokumentin jäädetyttyyn versioon.

5.2.1 eMon yliluokka

eMo:n yliluokka (questiontype.php) kommunikoi Moodlen kanssa käyttäen liitännäisrajapintaa. Alkuperäinen suunnitelma oli, että kysymystyypit kommunikoisivat Moodlen kanssa eMon yliluokan kautta, mutta toteutusvaiheessa huomattiin tällaisen ratkaisun olevan mahdoton. Siksi kysymystyypit kommunikoivat toteutetussa versiossa suoraan Moodlen kanssa ja eMon yliluokka tarjoaa vain kysymystyyppin lisäämis- ja poistopalvelut. Kuvaukset näistä komponenteista löytyvät suunnitteludokumentin lopullisesta versiosta.

5.2.2 Tehtävätyypin lisääjä

Luo tehtävätyypin nimeä vastaavan hakemiston question/type-hakemiston alle ja kopioi uuden tehtävätyypin tiedostot kyseiseen hakemistoon ja nimeää ne oikein. Aikaisemmassa suunnitelmassa hakemisto luotiin question/type/emo-hakemiston alle.

5.2.3 Tehtävätyypin poistaja

eMo-etusivu tarjoaa välilehden eMo-tehtävätyyppien poistamiseen. Välilehdellä näytetään lista tehtävätyypeistä ja listan alla poistopainike. Listan näyttää showremovetype.php, joka myös huolehtii siitä, että kysymystyyppejä, joista on olemassa kysymyksiä, ei voi poistaa. Tarkastus tehdään kyselemällä tietokannasta kaikki kysymystyypit, joista on kysymyksiä ja vertaamalla niiden listaa poistettavaan tehtävätyyppiin. Jos tehtäviä on olemassa, näytetään virheilmoitus eikä tehtävätyyppejä poisteta.

Koska toteutetussa versiossa eMo-tyypit käyttäytyvät kuten tavalliset kysymystyypit, tarkastetaan tyyppin kuuluminen eMo-järjestelmään etsimällä tyyppin hakemistosta emo.type-nimistä tiedostoa. Emo-järjestelmän avulla ei siis voi poistaa Moodlen oletustehtävätyyppejä, vaan vain eMon kautta.

5.2.4 Käyttöliittymä

eMon etusivulta on poistettu kysymyslista ja siellä näkyy ainoastaan ohjelinkki. Muihinkin käyttöliittymän osiin on toteutusvaiheessa tullut pieniä muutoksia.

5.2.5 Tehtävätyyppien tiedostokuvaus

Tehtävätyyppien questiontype.php tiedosto ei ole emo/type luokan aliluokka vaan default_questiontype -luokan aliluokka

5.2.6 Lokalisointi

Tämä tehdään emo-tyyppisissä tehtävissä antamalla erillisen lang-hakemiston kielikoodikohtaisissa (esim. en_utf8 tai fi_utf8) hakemistoissa tiedosto qtype_tyyppinimi.php, joka sisältää rivejä, kuten \$string['answerhowmany'] = 'One or multiple answers?'; Tehtävätyypin lisääjä kopioi nämä rivit Moodlen yleiseen lokalisointihakemistoon.

5.2.7 Kieliriippumattomuus

Moodlessa valmiina oleva Multilang-filtteri mahdollistaa palaute- ja kysymystekstien kääntämisen eri kielille helpolla XML-tyyppisellä koodauksella, mutta valitettavasti kysymysten nimiin se ei vaikuta. Se ei vaikuta myöskään kysymysteksteihin. Siksi valmiin templatepaketin mukana tulee emo_lang_filter.php, joka hoitaa tämän XML-koodauksen käsittelyn. Tätä ratkaisua kannattaa käyttää myös muissa tehtävätyypeissä, jotka vaativat monikielisyyden tukea.

5.2.8 SQL-luontilauseet

Käytettiin Moodlessa vaadittua XML-syntaksista SQL-kieltä.

```
<?xml version="1.0" encoding="UTF-8" ?>

<XMLDB PATH="question/type/emo/db" VERSION="20061209" COMMENT="Tables of the eMo-question plugin"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:noNamespaceSchemaLocation="../../../../lib/xmldb/xmldb.xsd"

>

<TABLES>

    <TABLE NAME="question_emo" COMMENT="question_emo table" NEXT="question_emo_options">

        <FIELDS>

            <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true" SEQUENCE="true"
ENUM="false" NEXT="question"/>

            <FIELD NAME="question" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="options"/>

            <FIELD NAME="options" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE="false" ENUM="false"
PREVIOUS="question"/>

        </FIELDS>

        <KEYS>

            <KEY NAME="primary" TYPE="primary" FIELDS="id" NEXT="question"/>

        </KEYS>

    </TABLE>

</TABLES>
```



```

        <KEY NAME="question" TYPE="foreign" FIELDS="question" REFTABLE="question" REFFIELDS="id"
PREVIOUS="primary"/>

    </KEYS>

</TABLE>

<TABLE NAME="question_emo_options" COMMENT="question_emo_options table" PREVIOUS="question_emo"
NEXT="question_emo_history">

    <FIELDS>

        <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true" SEQUENCE="true"
ENUM="false" NEXT="emoquestion"/>

        <FIELD NAME="emoquestion" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="name"/>

        <FIELD NAME="name" TYPE="char" LENGTH="100" NOTNULL="false" SEQUENCE="false" ENUM="false"
PREVIOUS="emoquestion" NEXT="value"/>

        <FIELD NAME="value" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE="false" ENUM="false"
PREVIOUS="name" NEXT="info"/>

        <FIELD NAME="info" TYPE="char" NOTNULL="false" SEQUENCE="false" ENUM="false"
PREVIOUS="value"/>

    </FIELDS>

    <KEYS>

        <KEY NAME="primary" TYPE="primary" FIELDS="id" NEXT="emoquestion"/>

        <KEY NAME="emoquestion" TYPE="foreign" FIELDS="emoquestion" REFTABLE="question_emo"
REFFIELDS="id" PREVIOUS="primary"/>

    </KEYS>

</TABLE>

<TABLE NAME="question_emo_history" COMMENT="question_emo_history table"
PREVIOUS="question_emo_options">

    <FIELDS>

        <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true" SEQUENCE="true"
ENUM="false" NEXT="question"/>

        <FIELD NAME="question" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="user"/>

        <FIELD NAME="user" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true" SEQUENCE="false"
ENUM="false" PREVIOUS="question" NEXT="response"/>

        <FIELD NAME="response" TYPE="char" LENGTH="1000" NOTNULL="false" SEQUENCE="false"
ENUM="false" PREVIOUS="user" NEXT="timestamp"/>

        <FIELD NAME="timestamp" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE="false"
ENUM="false" COMMENT="Default comment for the field, please edit me" PREVIOUS="response"/>

    </FIELDS>

```

```
<KEYS>

  <KEY NAME="primary" TYPE="primary" FIELDS="id" NEXT="question"/>

  <KEY NAME="question" TYPE="foreign" FIELDS="question" REFTABLE="question" REFFIELDS="id"
PREVIOUS="primary" NEXT="user"/>

  <KEY NAME="user" TYPE="foreign" FIELDS="user" REFTABLE="user" REFFIELDS="id"
PREVIOUS="question"/>

</KEYS>

</TABLE>

</TABLES>

</XMLDB>
```

6 Toteutumattomat vaatimukset ja suunnitelman osat

Kuten testausdokumentissa on todettu, kaikki vaatimukset on toteutettu. Vaatimuksesta J2, arvotut tehtävän muuttujat, ei ole olemassa esimerkkitoteutusta, mutta suunnitteludokumentissa ja käyttöohjeessa on annettu malliratkaisu, jonka avulla toiminto on mahdollista tehdä ja ohjelmistossa on tuki tälle toiminnolle.

7 Koodin ylläpitoon liittyvät seikat

Koodissa on pyritty käyttämään mahdollisimman paljon Moodlen valmiita kirjastofunktioita. Koska Moodle kehittyi edelleen nopeasti, joidenkin funktioiden toiminta voi muuttua, tai niitä voidaan alkaa pitää vanhentuneina. Tällöin näitä funktiokutsuja on muutettava.

Erityisesti mahdollisesti valmistumassa oleva formslib-kirjasto saattaa aiheuttaa järjestelmään muutoksia lomakkeen käsittelyn osalta tulevaisuudessa.

eMossa käytettyjä yhteisiä kirjastofunktioita on kerätty tiedostoon emofunctions.php, joka otetaan käyttöön useimpien projektin tekemien kooditiedostojen alussa require_once-lauseella. Tiedoston sisältämät funktiot tekevät lähinnä tiedosto- ja hakemistorakenteiden muutoksia. Tämä koodi voi olla hieman monimutkaista useiden silmukoiden ja tarkistuksien vuoksi, siksi sen siirtäminen omaan tiedostoonsa yksinkertaistaa muuta koodia, jossa tarvitsee käyttää vain emofunctions.php:n rajapintoja.

Yhtenä vaatimuksena oli erillinen laatija-rooli, jolla on oikeus lisätä ja poistaa tehtävätyyppejä. Tämä on toteutettu lisäämällä Moodleen oikeus ”mod/emo:create”. Laatija-roolin tekeminen ja tuon oikeuden antaminen sille on kuvattu käyttöohjeessa. Käyttöoikeus tarkastetaan useimpien eMon PHP-tiedostojen alussa Moodlen suosittamalla tavalla, eli tämäntyyppisillä koodiriveillä:

```
$context = get_context_instance(CONTEXT_MODULE);  
require_capability('mod/emo:create', $context);
```

Ensin \$context-muuttujaan haetaan nykyinen moduulikonteksti ja sitten tarkastetaan, että siinä on vaadittu oikeus. Require_capability-funktio on ns. assert-tyyppinen, eli jos oikeus on sopiva, tiedoston suoritusta jatketaan normaalisti. Jos oikeus ei ole riittävä, näytetään virheilmoitus ja lopetetaan tiedoston suoritus heti.

Lomakkeen käsittelyn turvallisuutta on lisätty käyttämällä jokaisessa lomakedataa käsittelevässä tiedostossa sessioavaimen varmennusta. Tällöin lomakkeen mukana lähetetään piilotettu sesskey-kenttä, jonka arvoksi annetaan PHP-funktion sesskey arvo. Lomakedataa käsittelevä tiedosto tarkastaa, että lähetetty sesskey ja tämänhetkinen sesskey ovat samoja funktiokutsulla confirm_sesskey(). Sen ympärille on tehty if-lause, joka tulostaa virheilmoituksen, jos avaimet eivät ole samat ja muussa tapauksessa jatkaa toimintaa normaalisti.

Jos eMo-järjestelmän tuottaman template-paketin sisältämiä tiedostoja halutaan muuttaa, täytyy muutokset tehdä hakemistoon emo/template/qtypename. eMo ottaa tästä hakemistosta tiedostot, jotka se pakkaa zip-pakettiin ja antaa käyttäjälle. Jos halutaan, että eMo korvaa myös uusissa tiedostoissa olevat ”oletustiedot” annetuilla tiedoilla, täytyy uusien tiedostojen nimet lisätä createtemplate.php-tiedostossa määriteltyyn \$listoffilestogothrough-taulukkoon.

8 Muut ylläpitoon vaikuttavat seikat

Moodlen uudet versiot voivat vaikuttaa eMo-järjestelmään. Tuote on tehty toimivaksi vain versiossa 1.7. Koska Moodle muuttuu niin nopeasti ja paljon, ei ole mahdollista taata tuotteen toimivuutta uudemmissa tai edes vanhemmissa versioissa.