

Peppi projekti: Selvitys lukujärjestyksien suunnittelutyökaluista

Versio 0.9
17.6.2011

Sisältö

1. Johdanto	1
1.1 Tavoitteet.....	1
1.2 Rajaukset.....	1
1.3 Taustaa.....	1
2. Työ- ja lukujärjestysuunnittelutyökalut.....	2
2.1 GP-Untis.....	2
2.2 Uni-Time 3.2	3
2.3 Time-edit	5
2.4 Scientia Syllabus Enterprise + Web Room booking ja Syllabus Plus.....	7
2.5 Web2Cal käyttöliittymäkirjasto	10
2.6 Asio-Data - lukujärjestysohjelmisto.....	11
3. Johtopäätökset kokonaisuudesta	11
4. Jatkotoimenpiteet	12

1. Johdanto

Tämä selvitys on tehty osana Metropolia ammattikorkeakoulun ja Tampereen ammattikorkeakoulun Peppi-projektia. Selvityksen pääpaino on selvittää olemassaolevien valmiiden lukujärjestystyökalujen teknistä sopivuutta Peppi kokonaisuuden vaatimuksiin. Tarkemmin sanottuna tämä tarkoittaa, että selvityksessä pyritään saamaan selville onko mikään järjestelmistä rakennettu palvelupohjaisen arkkitehtuurin mukaisesti tai onko järjestelmässä olemassa rajapintoja ja miten ne ovat hyödynnettävissä.

Selvityksessä tutkitaan myös hyvin pintapuolisesti muutamien lukujärjestyssuunnittelutyökalun ominaisuuksia suhteessa Peppi-projektissa tehtyihin toiminnallisiin vaatimuksiin.

1.1 Tavoitteet

Selvityksen tavoitteena on tutkia, löytyykö markkinoilta valmis lukujärjestyssuunnittelutyökalu, joka vastaa projektissa asetettuihin teknisiin vaatimuksiin (mm. palvelupohjainen arkkitehtuuri, reaaliaikaisuus, tietojen synkroninen siirto) sekä loppukäyttäjien antamiin vaatimuksiin.

1.2 Rajaukset

Tämän selvityksen ulkopuolelle on jätetty tutkittavien järjestelmien toiminnallisuus ja sopivuus työ- ja lukujärjestysten suunnitteluun. Toiminnallisen selvityksen järjestelmistä suorittaa työ- ja lukujärjestysryhmän asiantuntijat. Tämä selvitys ei siis ota kantaa siihen, mikä järjestelmä olisi toiminnallisuksiensa puolesta paras tai hyvä järjestelmä, vaan sen tarkoituksena on selvittää miten eri järjestelmät ovat liitettävissä Peppi kokonaisuuteen sovellusarkkitehtuurinsa ja rajapintojensa puolesta.

Selvityksen kohteeksi valittiin vain niitä lukujärjestyksen suunnitteluvälineitä, joita on tiedossamme käytössä tai tulossa käyttöön suomalaisissa korkeakouluissa. Selvitykseen rajattiin vain siis seuraavat ohjelmistot:

- gp-Untis (Metropolia ja TAMK)
- Uni-Time (Oulun yliopisto *tulossa)
- TimeEdit (Hanken)
- Scientia Syllabus (Lappeenrannan teknillinen yliopisto)
- Web2Cal (Aikuiskoulutus)
- Asio-Data - Lukujärjestysohjelmisto

1.3 Taustaa

Lukujärjestyssuunnittelun lähtökohtana Peppi-projektissa on se, että opetussuunnittelu etenee kokonaisuudessaan prosessina OPS-suunnittelusta aina lukujärjestyksiin saakka. Tämä tarkoittaa sitä että ennen lukujärjestyssuunnittelua muissa suunnitteluvaiheissa syntyy valtavasti informaatiota, jotka toimivat lähtötietoina lukujärjestyssuunnittelulle. Idea koko Peppi-projektissa on se, että eri prosessivaiheissa syntyvä tieto kuljetetaan järjestelmän sisällä prosessin seuraavaan vaiheeseen. Käytännössä halutaan siis eroon tietojen kopioinnista tai uudelleenkirjaamisista. Näin saadaan paitsi tehostettua suunnitteluprosessin kulkua, koska viimeisin syötetty tieto on reaaliaikaisesti saatavilla ja se on haettu aina tiedon omistavasta

tietolähteestä. Arkkitehtuuri vähentää virheiden määrää sekä tietojen kahdentumista ja tiedon sykkronoinnin aiheuttamia haasteita.

Edellä kuvattu ajattelutapa toimii myös tämän selvityksen ohjenuorana. Tarkoitus on löytää rartkaisu, jolla edellisestä prosessinvaiheesta (vuosisuunnittelu) voidaan suoraan siirtyä lukujärjestyssuunnitteluun ja joka on helppokäyttöinen ja tukee asiantuntijatyöryhmässä esitettyjä vaatimuksia.

2. Työ- ja lukujärjestyssuunnittelutyökalut

2.1 GP-Untis

http://www.grupet.at/home_en.php

Yleistä:

Untis on tällä hetkellä käytössä sekä Metropoliasa että TAMKissa. Molemmissa korkeakouluissa Untiksella suunnitellaan vain osa lukujärjestyksistä (joskin melko merkittävä osa). Untis on client-server –tyyppinen ohjelmisto ja soveltuu erityisesti keskitetyn mallin suunnitteluun. Metropoliasa ja TAMKissa Untiksella suunnitellut tiedot viedään siirtotiedostoina tilanvarausjärjestelmiin josta ne julkaistaan opiskelijoille. Siirron jälkeen tilavarauksia ja lukujärjestystietoja ylläpidetään tilanvarausjärjestelmissä.

Teknisiä asioita:

Hyvät puolet:

- Voidaan käyttää monen käyttäjän toimesta samaan aikaan
- Voidaan käyttää MySQL kannan kautta
- Omaa käyttökokemusta

Huonot puolet::

- Ei tue teknisen määrittelyn ensimmäistä ehtoa, jossa palvelu/ järjestelmä tulisi olla palvelupohjainen
- Tietokanta ei kuvattu eikä toimittaja ole valmis toimittamaan tietokannan kuvausta.
- Ei relaatiotietokantaa.
- Rajapinnat csv-muodossa siirtotiedostoina
- Tietojen siirto eräajopohjaista - ei reaaliaikaista tiedonsiirtoa
- Windows-client käyttöliittymä joten ei istu käyttöliittymältään muihin PEPPIn palveluihin
- Ei avoimen lähdekoodin järjestelmä eikä muutosten teko tai räätälöinti ole täten mahdollisia
- Tekninen dokumentaatio sekä tuen saaminen heikkoa

Toiminnallisuus:

Hyvät puolet:

- Tukee melko hyvin toiminnoiltaan vaatimusmäärittelyn näkymiä ja sijoittelua
- Monipuolisia toimintoja
- Sekä Metropoliasa että TAMKissa ollut käytössä joten osaamista on muutamilla henkilöillä

Huonot puolet:

- Ei sovellu satunnaiseen käyttöön tai hajautettuun malliin (vaatii paljon perehdytystä).
- Rajapinta puutteiden vuoksi virheiden syntyminen (koodit yms) yleistä.

- Ei tue täysin hajautettu mallia, missä opettajat pystyisivät tekemään muutoksia itse.

Hinta:

Yhteensä n. 5000€ vuodessa, jos organisaatiolla on jo järjestelmä käytössä.

Yhteenveto

Käyttäjätöiminnoiltaan varsin hyvä järjestelmä ja pitkän kehityksen tulos. Toisaalta kehitystä on tehty aiemman tekstipohjaisen järjestelmän kautta eli tietokantaan tallennetaan viittauksia samassa muodossa kun ne on tallennettu tekstitiedostoon ja tietokannan toiminnan selvittäminen on erittäin haasteellista (ei relaatioita tms.). Untiksen käyttöliittymä ei ole verkkopalvelupohjainen, joka helpottaisi useiden näkymien käyttöä. Näkymiin ei voida millään tavalla vaikuttaa eli Untikseen koodatut näkymät ja kentät ovat "kiveen hakattuja" ja niitä ei voida muokata. Untis ei tue kaikkia määrityksen mukaisia tietomalleja ja koska käyttöliittymää ei voida muokata, tämä aiheuttaa sen että Peppi:n muista palveluista tuotettua tietoa voitaisiin hyödyntää hyvin minimaalisesti. Muilla palveluilla tarkoitetaan esimerkiksi opettajan esittämiä toiveita opetuksen ryhtimykseen suhteen sekä toteutukselle kirjattuja tietoja pienryhmien määrästä jne. Täten esim. opettajan kirjaamaa tietoa ei voi siirtää Untis järjestelmään, vaan tiedot olisi jatkossakin haettava raporteina Peppi järjestelmästä ja tiedot tulee kirjata käsin Untikseen.

Ohjelma toimii Client-server -periaatteella. Ohjelmisto ei ole palvelupohjainen järjestelmä. Tietojen tuominen ja vieminen järjestelmään on toteutettavissa vain erä-ajoina, eikä eräajoja voi suorittaa suunnittelemattomasti. Joissain tapauksissa eräajo tuotantojärjestelmiin voidaan tehdä vain kerran ja virheitä ei saisi siirrossa tapahtua.

Untis ei sovellu tällaisenaan Peppin mukaisiin määrittelyihin mm. siksi että

- Suunnitteluprosessissa aiemmin tuotettua tietoa ei voida tuoda reaaliaikaisesti tähän järjestelmään Ei tue palvelupohjaisuutta
- Ei kunnollisia rajapintoja (vain tekstitiedostot)
- Ei dokumentoitua tietokantaa
- Ei tue hajautettua suunnittelumallia
- Ei keskitettyä käyttäjähallintaa

2.2 Uni-Time 3.2

http://www.unitime.org/unitime_intro.php

Tulossa käyttöön Oulun Yliopistolle (saamiemme tietojen mukaan)

Tekniset asiat

Hyvät puolet:

- Open source pohjainen ja siten muokattavissa omiin tarpeisiin.
- MySQL-pohjainen.
- Monta samanaikaista käyttäjää.
- Järjestelmän tekninen dokumentaatio on kattava.
- Tietojen siirto järjestelmään ja järjestelmästä ulos on huomioitu
 - XML siirtotiedostot

Huonot puolet:

- Ei lokalisoitua, saatavilla vain englanninkielisenä.

- Vain eräajotyyppinen tiedonsiirto
- Ei palvelupohjainen

Toiminnallisuudet:

Plussat

- Voidaan käyttää verkkopalveluna (web-käyttöliittymä)
- Hyviä yksittäisiä palveluja toteutettuna web-käyttöliittymän kautta
- Valmiina esim. rytmitysominaisuuksia joita voitaisiin tuoda PEPPIn muista palveluista (esim. kurssin X luennot 45min mittaisina 3 kertaa viikossa)

Miinukset

- Ei kovin intuitiivinen, ruma ja sekava käyttöliittymä
- Tehty yliopistokäyttöön ja ei tue ryhmien käyttöä ja käsittelee aineistoa kursseina
- Perustuu rajoitteiden asettamiseen ja itse sijoittelu tapahtuu automaattisesti annettujen rajoitteiden puitteissa. Ei tue riittävästi manuaalista sijoittelua.

Hinta

- Hintaa ilmainen ao. ehtojen mukaan:
- Constraint Solver moduuli on GNU LGPL-lisenssin alainen [GNU LGPL license](#), ja
- varsinainen sovellus on GNU GPL-lisenssin alainen University Course Timetabling Application under [GNU GPL license](#).

Yhteenveto

Suunniteltu yliopistokäyttöön ja periaatteena on se, että tehdään OPSeille lukujärjestykset siten että kurssit eivät menisi päällekkäin. Järjestelmä ei siis suoraan tue nykyisin AMKeissa käytettävää ryhmämallia ja opintojakso-opetuksen toteutus mallia. Muutoinkin käytöltään poikkeava, koska tässä järjestelmässä on keskitytty ratkaisumoottoriin, joka luo annettujen pohjatietojen perusteella työjärjestyksen, missä päällekkäisyyksien määrä on minimoitu. Ratkaisumoottorin tapaista vaatimusta ei tullut Pepissä esiin, eikä sen käyttöä ole määritely.

Järjestelmä pystyy myös huomioimaan kampuksien väliset välimatkat ja suunnittelun yhteydessä voidaan määrittää kuinka monta siirtymää kampukselta toisella sallitaan esimerkiksi yhden päivän aikana.

Käyttöliittymä on hankalakäyttöinen (kirjoittajan mielipide demoversion peusteella) ja muutenkin se on suunniteltu siitä lähtökohdasta, että käyttäjä syöttää vain rajoitteita ja järjestelmä itsessään optimoi lukujärjestykset ehtojen perusteella. Optimointi ei ole PEPPIn vaatimusten mukainen ominaisuus vaan suunnittelu ja sijoittelu halutaan jättää käyttäjille. Järjestelmä ei myöskään tue versiointeja ja hajautetun mallin suunnittelu on tällä järjestelmällä erittäin hankalaa jos ei mahdotonta.

Saamamme tiedon mukaan Oulun yliopisto on ottamassa tämän järjestelmän käyttöön lähitulevaisuudessa ja asiaa voitaneen selvittää sieltä tulevaisuudessa mikäli tarpeellista.

Uni-Time ei siis sovellu tällaisenaan Peppin mukaisiin määrittelyihin mm. siksi että

- käyttöliittymä kaukana vaatimusmäärittelyistä
- käyttö suunniteltu optimointitoimintoa ajatellen

- käyttöliittymä ei saatavissa suomeksi (onnistuisi varmaan lisätyönä)
- riittämättömät sijoittelunäkymät
- vaatisi joka tapauksessa paljon kustomointia/räätälöintiä

2.3 Time-edit

<http://www.evolvera.se/htmlinternational/eng/eng.timeedit.university.html>

Käytössä mm. Hankenilla

Tiedot perustuvat verkosta saatavaan tietoon sekä Hankenillä tehtyihin haastatteluihin. Osa tiedoista perustuu vanhaan 1.4(?) versioon ja osa uuden 3. version luvattuihin ominaisuuksiin (ei käytössä Suomessa).

Miten Hankenilla toimitaan

- Keskitetty suunnittelu (yksi tekee koko koululle)
- Vaasan ja Helsingin toimipisteiden lukujärjestykset suunnitellaan täysin erikseen
- Muutoksista ilmoitetaan sähköpostilla tms. - ei automatiikkaa, ei yleensä poisteta merkintöjä lukkarista vaikka opettaja sairastuu
- Kurssinimet/tunnukset tuodaan siirtotietona mutta koska tunnukset säilyvät samana niin käytännössä tuodaan vain uudet kurssit - mitenköhän Metropolia/TAMK tapauksissa kun toteutuksia tulee todella paljon

Tekniset asiat

Hyvät puolet

- Vanha versio 4d-tietokantaan perustuva (melko eksoottinen) - Uusi 3. versio ilmeisesti MySQL pohjainen
- 3. versio SaaS -pohjainen järjestelmä
- Web-service -rajapintoja mahdollista toteuttaa itse kunhan käy ensin Ruotsissa koulutuksessa

Huonot puolet

- Rajapinnat heikot ainakin ennen versiota 3. (uudesta ei varmaa tietoa millaiset rajapinnat)
- Vaikkakin järjestelmään voidaan viedä ja tuoda dataa rajapintojen avulla ei se toimi palvelupohjaisesti
- Vain englanniksi, norjaksi, saksaksi ja ruotsiksi

Toiminnallisuudet

Hyvät puolet

- Koko version jos ottaa niin mukana tulee tilanvarausjärjestelmä ja raportointijärjestelmä
- Autentikointi onnistuu LDAPin kautta (CAS:sta ei tietoa)
- Varauksien tekeminen onnistuu suht. helposti eli hiirellä klikataan kalenterista oikeaa kohtaa ja täytetään avautuva lomake
- Toistuvat varaukset ym. onnistuvat
- Ulkopuolisen tilan varaukset ovat mahdollisia
- Kurssit/toteutukset voidaan kategorisoida melko hyvin eri tasoihin (klusteri->koulutusohjelma)
- Tilat voidaan kategorisoida esim. toimipisteittäin

- Pystytään luomaan ryhmiä (tukee siis ryhmäkäsitetä) - käytössä ehkä selviää tukeeko saapumisryhmä/pienryhmä -käsitteitä. Hankenilta jäi kuva että kyseessä on nimenomaan pienryhmätoiminto
- 3. versiossa näyttäisi toimivan ikkunointi melko hyvin eli voisi olla monta näkymää samaan aikaan auki
- Varaukset on mahdollista asettaa "to-be-released" -tilaan eli luonnokseksi
- Kohtuullisen hyvät sijoittelunäkymät
- Näkymien yhdistäminen (eri resursseja esim. samaan viikkonäkymään) onnistuu

Huonot puolet

- Varauksien tekemisessä resurssien kiinnittäminen tapahtuu ilmeisesti aina varausvaiheessa eli ainakaan Hankenilla niitä ei pystytä tuomaan suoraan aiemman suunnittelun perusteella automaattisesti
- Mallipohjakäsitettä ei ole sellaisenaan

Hinta

- Pari vuotta vanha hinta-arvio 3. versiolle ~5€ per opiskelija/henkilökunta per vuosi

Muuta:

Esimerkki Time-editin varaustiedoista;

Day: Tue, Jan 11, 2011

Time: 10:15-11:45

Week: 2

Group: KK3 K 3
SK3 S 3
APK3 Allmän pedagogik 3
VuxP3 Vux.ped åk 3

Course: ITarb IT som arbetsredskap

Teacher: AndSu Andreas Sundstedt

Room: C201 C201 Auditorium Bruhn

[Last modified on 10/29/2010 at 11:14]

Yhteenveto

Käyttöliittymä vaikuttaa suhteellisen helppokäyttöiseltä ja mahdollistaa usean resurssin päällekkäisen näkymisen. Kokemuksia vain yhden keskitetyn käyttäjän toimintamallista ja tutustumisen pohjalta ei käynyt selväksi voisiko tuo toimia miten hyvin monen yhtäaikaisen käyttäjän näkökulmasta. Lisäksi ollut käytössä vain yhdessä kiinteistössä ja ei ole varmuutta miten järjestelmällä onnistuu ison korkeakoulun kiinteistöjen hallitseminen. Muutenkin tätä käytetään vain suunnitteluun ja esim. muutoksia ei korjata yleensä järjestelmään.

Rajapinnat iso kysymysmerkki eli Hankenin mukaan siinä asiassa on paljon parannettavaa. Jotain kuitenkin odotetaan rajapintojen osalta paranevan uuden version myötä.

Pieni kysymysmerkki on, miten hyvin yhteistyö onnistuu toimittajan kanssa, koska järjestelmä vaatisi joka tapauksessa paljon muutoksia, jotta päästäisiin edes lähelle PEPPI-määrityksiä. Käytännössä tämä järjestelmä vaikuttaa melko normaalilta tilanvarausjärjestelmältä johon on lisätty usean resurssin päällekkäisnäkyä ja melko hyvin toteutettu käyttöliittymä.

Vanhaan TimeEdit järjestelmään oli saatavissa rajapintoja, joiden kautta tietoa voidaan lukea TimeEdit tietovarannosta. Uusi TimeEdit järjestelmä on kokonaan SaaS pohjainen ja tämä asettaa tiettyjä haasteita mm. rajapintojen rakentamiseen ja/tai järjestelmän räätälöintiin Metropolian ja TAMK:n tarpeisiin. Jos järjestelmä kytketään osaksi Peppiä ja järjestelmien välillä tehdään hyvin paljon tietokantahakuja ja rajapintoja on riskinä vasteajan kasvaminen, koska osa tiedoista haetaan Ruotsissa olevilta palvelimilta.

Uusi TimeEdit tarvitsisi lisäselvitystä, jotta voitaisiin sanoa, onko rajapinnat riittävät Pepin tarpeisiin. Haasteena kuitenkin on se, että *uutta* TimeEdit järjestelmää ei ole käytössä missään suomalaisessa korkeakoulussa.

Time Edit ei siis sovellu tällaisenaan Peppin mukaisiin määrittelyihin mm. siksi että

- suunnitteluprosessissa aiemmin tuotettua tietoa ei voida tuoda reaaliaikaisesti tähän järjestelmään (uuden version osalta vielä selvitettävä)
- ei tue hajautettua suunnittelumallia kovinkaan hyvin
- suomenkielisen käyttöliittymän puute
- SaaS palvelun asettamat haasteet integraatioiden ja nopeuden suhteen
- uuden version rajapintamahdollisuudet ja palvelupohjaisuus "promisewarea" eli vaatii selvittelyitä mikäli tällä halutaan jatkaa

2.4 Scientia Syllabus Enterprise + Web Room booking ja Syllabus Plus

Tiedot perustuvat osittain esitteisiin (osa vuodelta 2008) ja osittain Lappeenrannassa tehtyyn haastatteluun

<http://www.scientia.com/uk/>

Käytössä Lappeenrannan teknillisellä yliopistolla (SyllabusPlus)

Idea se, että varsinainen suunnittelujärjestelmä on tuo Syllabus Plus ja sen päälle tulee tuo web Room booking ad-hoc - ja hajautettua suunnittelua varten (Scientian verkkosivujen mukaisesti).

Lappeenrannan prosessi lyhyesti. Lukujärjestykset suunnitellaan lukuvuosittain. Suunnittelu alkaa vuodenvaihteessa ja opettajille tehdään lomake, jossa he ilmoittavat jaksotukset ja muut toiveet omille kursseilleen. Tämän jälkeen lukujärjestyksien rajoitteet ja toiveet tallennetaan järjestelmään ja tehdään automaattinen sijoittelu. Tämän jälkeen syntyneet lukkarit tarkastetaan sekä suunnittelijan että opettajan toimesta. Lukujärjestykset lähetetään opettajalle sähköisesti esim. pdf:nä. Lukujärjestykset ovat valmiita koko seuraavalle lukuvuodelle ennen lukuvuotta edeltävää kesää. Varsinioden suunnitteluun menee aikaa noin kaksi päivää. Tätä kuitenkin edeltää rajoitteiden syöttäminen. Lappeenrannassa pohjana käytetään edellisen vuoden lukujärjestyksiä ja lukujärjestykset tehdään koulutusohjelmittain. Käytännössä suunnittelu on vielä pilkottu siten että ensin suunnitellaan/sijoitellaan 1. vuoden opiskelijoiden lukkarit, sitten 2. vuoden opiskelijat jne..

Huomioita Lappeenrannasta

- Otettu käyttöön vuonna 2004
- Sovittu tiettyjä käytäntöjä esim pidetään yleensä aina 2h tunteja
- Suunnittelu koordinoitu koko koulun tasolla eli yhteensä 13 henkeä (koulutusaloittain) tekee lukkarit ja yksi on koordinoija jotta aikataulu pysyy hanskassa.

- Noin 750 kurssia suunnitellaan ja sijoitellaan per vuosi, kurssien nimet eivät vaihdu kovin tiuhaan vaikka sisältö saattaakin muuttua. Tämä helpottaa lukkarinsuunnittelua koska voidaan usein peilata edelliseen vuoteen. Kurssitiedot päivitetään käsin Syllabukseen.
- Syllabus käytössä Suomessa ilmeisesti vain Lappeenrannassa
- Lappeenrannassa lähes kaikki varaukset tehdään Syllabuksella mutta esim. neukkareita varten on eri systeemi. Lukkarit julkaistaan oodin kautta ja muutokset tulevat myös sinne (ei siis päivity tilavarauksiin!)

Ominaisuuksia:

- Järjestelmä periaatteessa keskitetyn suunniteelun järjestelmä mutta päälle saa hankittua web-varausjärjestelmän ad-hoc varauksia varten. Lappeenrannassa tosin ei erillistä web-varausjärjestelmää vaan menee lomakkeella muutospyyntönä jonka suunnittelija/virkailija korjaa järjestelmään
- Speksien mukaan tukee sekä hajautettua että keskitettyä suunnittelumallia, mutta käytännössä tällä tarkoitetaan resurssien omistajuutta järjestelmätasolla - ei niinkään käytettävyyttä. Lappeenrannassa käytössä vain keskitetty suunnittelu.
- Tiloja pystyy merkitsemään kaikkien varattaviksi tai roolin mukaan esim. koulutusohjelman varattaviksi
- Oracle tai MS SQL - tietokannat valittavissa kumpaa käyttää
- Joka vuodelle on luotava uusi tyhjä tietokanta eli 1 suunnittelujakso / 1 uusi tietokanta lähtötietoineen
- Web-room booking on ASP.NET -pohjainen (tätä ominaisuutta ei ilmeisesti käytössä Lappeenrannassa)

Käytettävyys

Plussat

- Web-varauksissa näyttäisi olevan kohtuullisen hyvät hakuvaihtoehdot
- Lukuisia eri näkymiä (Timetabler -client)
- Tukee usean resurssin yhtäaikaista näkymiä
- Voi tehdä toistuvia varauksia esim. periodin ajaksi tai vastaavasti käydä muuttamassa jonkun yksittäisen varauksen tietoja
- Varauksia voi tehdä ulkopuoliseen tilaan lisätietokentän avulla (tieto lpr)
- Undo-Redo -napit eli järjestelmä osaa jäljittää muutokset
- Käyttäjä pystyy personoimaan oman client-näkymänsä (muokattavuuden taso ei täysin selvinnyt, joitain nappeja voi esim. piilottaa jne.)
- Tukee opiskelijaryhmiä (Student set), ei tietoa voiko näitä tuoda jostain
- Tukee mallipohja-ajattelua (activity), niitä voi myös kopioida.
- Tietokannasta voi tallentaa itselleen snapshot-imagena ja jatkaa työskentelyä kotona. Kun järjestelmä taas kytketään kantaan niin järjestelmä tarkastaa päällekkäisyydet/virheet

Miinukset

- Syllabus Plus käyttöliittymä todella vanhanaikaisen ja sekavan näköinen, Lappeenrannan kokemusten mukaan käytön opettelu kunnolla vaatii vuoden harjoittelun
- Opetuksen jaksotustoiveille ei löydy paikkaa (Lappeenrannassa erilliset formitt tätä varten)
- Ei versiointia
- Kaksi mallipohjaa voidaan liittää samaan tilaan eli kaksi toteutusta voidaan laittaa samaan luokkatilaan

- Useampi opettaja voidaan liittää samaan varaukseen
- Ryhmätiedot kirjoitetaan käsin tai generoituu automaattisesti (epäselvää voiko näitä tuoda jostain)
- Lappeenranta-demon perusteella todella hankala arvioida miten sujuu Metropoliassa ja Tamkissa yleisesti käytetty suunnittelumalli jossa ei käytetä automaattista sijoittelua vaan kaikki tehdään käsin.
- Ei sovellu hajautettuun suunnittelumalliin

Tekniset asiat

Plussat

- Raporttitietokannasta sanotaan näin: "SQL Server and Oracle versions are available". Lappeenrannasta saatujen tietojen mukaan Syllabus Enterprise-versiossa tulee myös tietokantakuvaukset.
- Käyttöliittymä kustomoitaviss osittain web room booking -versiossa (formit, värit)
- LDAP autentikaatio, Windows autentikaatio (web-varaukset ainakin). Näin siis Syllabuksen sivujen mukaan, Lappeenrannan haastatteluissa kävi ilmi että käyttäjähallinta on tehtävä Syllabuksessa itsessään.
- Web-varausjärjestelmä (ja Syllabus) tukee roolitusta eli osa tiloista voidaan piilottaa/näyttää roolin mukaan (esim. koulutusohjelma tms)
- Varausformeja pystytään ilmeisesti kustomoimaan (lisäämään poistamaan kenttiä lähde Scientian verkkosivut)
- Varausjärjestelmä tukee tilojen aukioloja periodi-/viikkotasolla (tilat varattavissa X aikana) - Ei tiedoa voidaanko tila asettaa varattavaksi vain tietylle kellonajalle
- Yksisuuntianen Outlook-yhteensopivuus "The timetable calendar can be viewed alongside or merged with the user's personal calendar (when using Microsoft Outlook) but activities in the timetable can not be rescheduled." Voi olla kyseessä ns. iCalendar -kalenteri eli oman kalenterin rinnalle voi tuoda ics. kalenterin.

Miinukset

- Speksien mukaan ilmoitukset varauksista käyttäjälle aina sähköpostilla joka screenshotien mukaan kirjoitetaan käsin! Näin myös Lappeenrannassa eli ei automaattisia ilmoituksia tai vastaavaa. Tiedot siirretään eräajona Oodiin josta ne menevät opiskelijoille.
- Ei ilmeisesti pystytä tuomaan reaaliaikaisia lähtötietoja muista järjestelmistä (näin ainakin vanhassa Syllabuksessa). Lappeenrannassa käytetään COM-rajapintaa eli tietoja kierrätetään Excelin kautta.

Hinta

Lisenssintimalli käyttäjämäärä per kanta (joka vuosi uusi kanta, tieto lpr)

Yhteenveto

Järjestelmää arvioitaessa menee helposti sekaisin SyllabusPlus joka on käytössä Lappeenrannassa sekä Enterprise-versio jonka tiedot pohjautuvat pääasiassa Scientian omiin esitteisiin.

Järjestelmä todennäköisesti potentiaalisin tutkituista valmispakettijärjestelmistä. Tämä ei kuitenkaan tarkoita että järjestelmä olisi sopiva Peppin mukaisiin määrityksiin. Järjestelmää ei ole suunniteltu palvelupohjaiseksi järjestelmäksi vaikka tietokantakuvaukset ja relaatiotietokanta Enterprise-versiossa saattavat mahdollistaa joidenkin palvelujen rakentamisen tämän järjestelmän päälle. Tietojen tuominen/vieminen on kuitenkin oletuksena tehtävä eräajoina.

Järjestelmän toiminnot ovat erittäin kattavia ja järjestelmä vaikuttaakin suunnitellun siten, että sitä on tarkoitus käyttää automaattisen sijoittelun periaatteella.

Yksi huomioitava seikka on tämän järjestelmän suhteutuminen muihin tilavarauksiin sekä lukujärjestyksien julkaisu. Lappeenrannassa lukkarit julkaistaan oodin kautta eli lukkarit eivät ole synkassa tilavarausten kanssa. Tämä vaikuttaa suuresti siihen miten järjestelmä olisi kytkettävissä Peppiin ja mikä olisiärkevin lukkareiden julkaisutapa ja mikä tilojen hallintajärjestelmä.

Syllabus ei siis sovellu tällaisenaan Peppin mukaisiin määrittelyihin mm. siksi että

- suunnitteluprosessissa aiemmin tuotettua tietoa ei voida tuoda reaaliaikaisesti tähän järjestelmään (näin ainakin syllabus Plus)
- se ei ole palvelupohjainen järjestelmä
- sitä ei ole saatavissa suomeksi
- se ei sovellu hajautettuun suunnitteluun

2.5 Web2Cal käyttöliittymäkirjasto

<http://www.web2cal.com/>

Kyseessä on valmis käyttöliittymäkirjasto, joka voidaan ottaa käyttöön esim. tilavaraus tai lukujärjestyksen suunnitteluväläneessä. Tämä tarkoittaa sitä, että toimintalogiikka, sovelluskerros ja tietovarannot joudutaan tekemään erikseen.

Tekniset asiat

Plussat

- Toimintalogiikka ja sovelluskerros rakennettavissa palvelupohjaiseksi ja rajapinnat määriteltävissä itse
- Tehty kustomoitavaksi
- Tukee palvelupohjaista ajattelumallia
- Valmiita käyttöliittymämoduuleita

Miinukset

- Vaatii paljon koodaamista ennen kuin käyttökelpoinen
- Vaatii kokonaan palveluiden ja toimintalogiikan rakentamisen

Toiminnallisuudet

Plussat

- Muistuttaa Google kalenteria eli käyttö helppoa
- Tukee drag&droppia ja on intuitiivinen
- Tukee useita näkymiä

Miinukset

- Ei sisällä valmiiksi kaikkia tarvittavia tietokenttiä kuten pienryhmät jne.
- Osa käyttöliittymä komponenteista, mitä Pepissä on määritelty, tulisi kehittää erikseen.

Hinta

Ultimate-versio \$1499 / vuosi

Yhteenveto

Web2Cal eroaa kaikista muista selvityksen ratkaisuista siinä, että Web2Cal ei varsinaisesti ole ohjelmisto vaan valmis käyttöliittymäkirjasto. Näin ollen Web2Cal tarjoaa työkalut kehittää oma lukujärjestyksien suunnittelutyökalu, mutta tilanvaraustieokannat, käyttölogiikat yms. tulee kehittää itse.

Web2Cal:in avulla on mahdollista toteuttaa täysin Peppi-projektin määritysten mukainen järjestelmä joskin se vaatii työtä. Web2Cal on suunniteltu nimenomaan tukemaan palvelupohjaista arkkitehtuuria ja on siksi erittäin potentiaalinen vaihtoehto.

Alustavasti Web2Cal soveltuu osin Peppin mukaisiin määrittelyihin mm. siksi että

- siitä voidaan tehdä palvelupohjainen
- se voidaan rakentaa olemassa olevien ja Peppissä tuotettavien ratkaisujen päälle
- siitä voidaan rakentaa välimuoto sisältäen osan nykyisten tilavarauksjärjestelmien ja markkinoilla olevien suurempien lukujärjestyssuunnittelutyökalujen ominaisuuksista
- MIINUS: vaatii suhteellisen paljon koodaamista eli ei missään tapauksessa valmis tai edes puolivalmis järjestelmä
- Vaatii lisäselvittelyitä toimittajan (Eduix) puolelta, kuinka paljon tämä vähentäisi käyttöliittymien tekemiseen arvioitua työmäärää.

2.6 Asio-Data - lukujärjestysohjelmisto

Tähän ohjelmistoon tutustutaan mahdollisesti mahdollisuuksien mukaan elo-syyskuun aikana tarkemmin.

3. Johtopäätökset kokonaisuudesta

Tutkituista ohjelmistoista yksikään ei tukenut täysin Peppi-projektin määrittelyä. Huomion arvoista on, että osa järjestelmistä vastaa kohtuullisen hyvin käyttöliittymälle asetettuja vaatimuksia mutta niiden suurin puute on tekninen yhteensopivuus muiden järjestelmien kanssa, sekä sovellusarkkitehtuuri, joka ei tue palvelukeskeisyyttä. Näihin tuotteisiin kuuluu mm. Untis, TimeEdit sekä Syllabus. Kaikki edellä mainitut ohjelmistot on suunniteltu keskitettyä suunnittelumallia silmällä pitäen, eivätkä ne sovellu hajautettuun suunnitteluun. Tällaisille ohjelmille on ominaista hyvin monipuoliset toiminnot, mitä tulee suunnittelun/sijoittelun toteuttamiseen ja automatisointiin mutta vastaavasti ohjelmiston opettelu vaatii suhteellisen paljon aikaa. Myös käyttäjähallinta on useimmissa tapauksissa hoidettava ohjelmassa itsessään. Tyypillistä ohjelmistoille on se, että tiedot syötetään niihin suunnittelijan toimesta ja vain harvat tiedot tulevat muista järjestelmistä. Jos tietoa pystytään tuomaan muista järjestelmistä ne tuodaan eräajopohjaisesti ja harvoin. Päivitykset tietojen osalta on helpompi tehdä käsin, jos muutettavia tietoja ei ole paljon. Tämä aiheuttaa tilanteita, joissa tiedot eivät ole sykkronissa eri järjestelmissä.

Mielenkiintoisen vaihtoehdon tarjoaa Web2Cal. joka ei ole varsinainen ohjelmisto vaan valmis

käyttöliittymäkirjasto, joka mahdollistaa käyttöliittymän integroinnin ja kehittämisen olemassa olevien järjestelmien päälle. Tämä voisi mahdollistaa kevyemmän suunnittelujärjestelmän rakentamisen hajautetun suunnittelumallin tueksi. Tämäkin tosin vaatii aikaa ja rahaa toteutuakseen eli ei ole ns. kaupasta ostettava ratkaisu.

Vastaavasti jos ajatellaan koko suunnittelujärjestelmän toteuttamista kokonaan itse, on riskinä projektin paisuminen kustannusten ja aikataulun osalta. Tutkituissa suunnittelujärjestelmissä on valtava määrä erilaisia ominaisuuksia ja optimointeja joita kaikkia ei kannata lähteä itse tekemään, mutta samalla voidaan kysyä ovatko kaikki toiminnallisuudet tarpeellisia? Haasteena onkin löytää "minimikokoonpano" ominaisuuksista, joilla keskitetyn suunnittelumallin toimijat pystyvät suunnittelemaan lukujärjestykset ja vastaavasti hajautetun mallin käyttäjät pystyvät omaksumaankin järjestelmän käytön.

Harkitsemisen arvoinen vaihtoehto lyhyellä tähtäimellä voisi olla myös se, että lähdetään tekemään kevyempää suunnittelujärjestelmää hajautetun mallin (sosiaali- ja terveysala sekä kluttuuriala) suunnittelijoiden käyttöön, koska tällä hetkellä heille ei ole tarjota minkäänlaista suunnittelutyökalua. Järjestelmän pohjana voisi olla jompi kumpi Metropolian tai TAMKin tilanvarausjärjestelmistä tai niiden yhdistelmä sekä käyttöliittymän pohjana esimerkiksi Web2Cal.

Tällä ratkaisulla ei päästäisi kokonaistavoitteeseen, jossa suunnitteluprosessi on kaikilla tasoilla yhtenäinen (jäisi edelleen esim. Untis-käyttö keskitetyn suunnittelun osalta) ja että käytössä olisi vain yksi järjestelmä, mutta olisi silti askel oikeaan suuntaan. Myöhemmin järjestelmää pystyttäisiin ehkä kehittämään eteenpäin, jolloin saataisiin tuettua paremmin myös keskitettyä suunnittelua.

Suunnittelujärjestelmävalintojen suhteen tulisi miettiä samalla sekä Metropolian että TAMKin nykyisten tilanvarausjärjestelmien kohtaloa. Tullaanko niitä liittämään osaksi tulevaa ratkaisua? Osa tässä selvityksessä esitetyistä suunnittelujärjestelmistä pitävät sisällään oman tilahallinnon sekä varausjärjestelmän. Tilanne, jossa suunnittelujärjestelmä ja tilanvarausjärjestelmä toimivat irrallisina järjestelminä saattaa aiheuttaa hankaluuksia varauksien ja tilojen synkronoinnissa. Mikä järjestelmä omistaa minkäkin tiedon ja mistä järjestelmästä esimerkiksi opiskelijan tai opettajan työjärjestykset katsotaan ja missä eri tietoja ylläpidetään?

Em mallissa tietoa joudutaan sykronoimaan eri tietolähteiden välillä, jotka omaavat oman tietomallinsa. Tietojen sykronoiminen eri tietolähteiden välillä on juuri se ongelma ja haaste, jota palvelupohjaisessa arkkitehtuurissa pyritään välttämään.

4. Jatkotoimenpiteet

Mikäli tämän selvityksen jälkeen ilmenee jatkoselvitystarpeita, ehdotamme, että mahdollinen selvitys keskittyy vain niihin ohjelmistoihin, jotka on mahdollista saada järkevästi toimimaan siinä teknologisessa ympäristössä, johon eri koulutuksen suunnittelujärjestelmät asettuvat. Tämä tarkoittaa sitä, että käsityksemme mukaan ei ole järkevää lähteä tutkimaan lisää niitä vaihtoehtoja, jotka voisivat toiminnallisilta ominaisuuksiltaan olla loistavia, mutta joihin suunnittelun aikana syntynyttä tietoa ei ole järkevästi siirrettävissä ja/tai joista tietoja ei ole saatavissa järkevästi esimerkiksi julkaisujen (tilanvaraus, sähköpostikalenteri, info-tv, mobiilisovellukset jne.) käyttöön.

Ehdotamme, että ohjausryhmä tarkentaa seuraavassa kokouksessaan, mitkä ovat ne tavoitteet, mihin lukujärjestyksen suunnitteluvälineellä pyritään.

- Riittävätkö nykyiset välineet ja palvelut niihin tarpeisiin ja tavoitteisiin, jotka organisaatiot ovat asettaneet?

- Jos eivät riitä, niin miltä osin erityisesti parannuksia pitäisi tulla ja missä laajuudessa?
- Minimitaso jolla pärjätään?
- Ohjausryhmän visio työ- ja lukujärjestyksen suunnitteluprosessista ja tuloksista?

Nämä tavoitteet ohjaavat mahdollisen lisäselvityksen tavoitteita sekä selvityksen sisältöä.