

25.9.2010
Arkkitehtuurikuvaus
Ohjelmistotuotantoprojekti kesä 2010
PAREMPI
University of Helsinki
<http://cs.helsinki.fi/group/ohtu/>

Based on "HP Architecture Template for Documenting Software and Firmware Architectures"
http://cs.helsinki.fi/group/kuje/misc/HP_arch_template_vers13_withexamples.pdf

1. Johdanto: yleinen kuvaus

Projektissa toteutetaan rendereri plastexiin käyttämällä mahdollisimman paljon hyväksi alkuperäisiä plastexin toiminnallisuuksia. Renderer luo .tex-tiedostosta .txt-tiedoston, joka on ihmiskäyttäjälle luettavuudeltaan .tex-tiedostoa parempi. Erityisesti tarkoituksena on tehdä pistenäytöltä helposti luettavissa oleva tekstitiedosto siten, että käyttäjälle kuitenkin jätetään mahdollisimman paljon työkaluja konfiguroida tekstitiedoston ulkoasua. Konfigurointimahdollisuuksia ovat esimerkiksi tekstin leveys, erinäisten kommentojen ulkoasu, tulostuksen erittely erillisiin tiedostoihin, tunnistamattomien kommentojen käsittely ja matemaattisten symbolien ulkoasu.

2. Vaatimukset

2.1 Toiminnalliset vaatimukset

Asiakas toimitti vaatimukset meille neljänä tekstitiedostona, jotka vielä 25.9.2010 löytyivät sivustolta <http://hlab.dy.fi/ml/>. Alla kuitenkin vielä kunkin tiedoston sisältö eriteltynä omiksi osikseen. Lopulliset vaatimukset sykleittäin eriteltynä olivat seuraavat:

Sykli 1:

- section/subsection...
 - tällä hetkellä osaa vain "toisen asteen" eli 1.1 subsectionit, mutta ei sitä pidemmälle (ei siis osaa 1.1.1)
- listat (itemize, enumerate)
 - sisennyksen muuttaminen (tehdään nyt välilyönteinä sekä tabilla ja laitetaan ini-tiedostoon inttinä; myöhemmin tarkastellaan jos lisätään tabulator mukaan myös jollain tavalla)
 - väli bullet pointin jälkeen konfattava
 - miten pistenäyttö esittää bulletpointin?
- konfigurointitiedosto

Sykli 2:

- dokumentointi monessa muodossaan

- defaultiksi konfiguraatio, joka ei hukkaa informaatiota (mm. korostus), aggressive tai passive
 - babel? (tkl-tiki-pohja avuksi muotoiluun/suomennoksiin joskus myöhemmin?)
 - node tutuksi
 - listan ominaisuuden muokkaaminen (bugi)
 - tekstin leveys (rajoittamaton rivinpituus), emph, lihavointi,
 - kappalejako konfiguroitavaksi: sisennys vai ei.
 - taulukot
 - selvitä mitä plastex nyt tekee taulukoille
 - aloitetaan tabular-ympäristöstä
 - hline ja vline -ominaisuudet
 - uusiokäytä array-funktiota
 - Sykli 3:
1. sisällysluettelo(toc-tiedosto)
 - a. tiedoston luku
 - b. sisällysluettelon kirjoittaminen
 2. kuvat
 - a. tiedoston nimi
 - b. sisennys
 - c. kuvateksti
 - d. includegraphics ylikirjoittaminen?
 3. quote(toimii jotenkin)
 - a. confattavaksi
 - b. sisennys pois
 4. taulukot (joskus ehkä sitten htb?)
 - a. hline
 - b. vline
 - c. rlc
 - d. ||
 - e. =
 - f. cline
 5. konfit (säännöllisiä lausekkeita?)
 - a. ei saisi hukata
 6. haxtex (tehty)
 -
 - Sykli 4:
- table-ympäristö
 - keskitys pois (konfiguroitavaksi)
 - sisällysluettelon sisennys
 - matikkaa (ascii math notaatio oletuksena)
 - debuggeri
 - esimerkki kurssisivulta: <http://www.cs.helsinki.fi/group/ohtu/ke-2010/latex.html>
 - label/ref (ilmeisesti toimii jotenkin?)
 - informaatiota hukkaamaton versio
 - ignorointi
 - tunnistamattomien kommentojen tulostus halutulla notaatiolla

- notaatio latexmainen: `\begin{komento} ... \end{komento}` jos vaan onnistuu
- sivut tekstitiedostossa, konfiguroitavaksi (vrt pdf), tavoitteena ensisijaisesti kappaletarkkuus
 - latexista lähtevä
 - labelien lisäys lähdekoodiin tarpeeksi tiheästi
 - latexin valmiiden paragraphien käyttö (http://en.wikibooks.org/wiki/LaTeX/Document_Structure#Sectioning_Commands)
 - pdf:n lukeminen
 - dvi:n lukeminen (dvitype wikipedia)
 - tietääkö plastex sivunumerot?
 - ulkonäkö:

----sivu 1----

.

.

.

----sivu 2----

- lähdeviitteet (lähdeviitteiden notaatio, huom. alaviitteet oletusarvoisesti samanlaisia plastexin käsittelyn jälkeen)
- alaviitteet kappaleen (paragraphin) perään
 - mites numerointi? kappaleen sisällä numeroituina vai koko dokumentin sisällä?
 - notaatio `{fn nro ...}`

Syklit 5:

- matematiikka
 - array-ympäristö matematiikassa
 - align-ympäristö
 - left ja right aaltosulkuina
 - displaymath
 - math (jos kappaleen ainoa lapsi --> poista dollarit)
 - konfiguroitavaksi jos ei ole kappaleen ainoa lapsi
 - text-ystö matikka-ystössä
 - konfit (ascii math / hlub)
 - matematiikan symbolit
- pageref (uudet nimet labeleille?)
 - konfiguroitavat nimet?
- kalvoympäristöt(perus slide generoidut numerot, beamer?)
- koodin refaktorointi

sykli 6: (LAMA-moniste, muista ajaa mps.tex)

- dokumentointi (pydoc, käyttöohjeet, ...)

- esimerkkejä
 - jatkokehitys: tuntemattoman komennon lisäys
 - tuntemattoman komennon tunnistus
 - tuntemattoman komennon toteutus
 - symbolien lisäys (.latextextsymbols)
 - komentojen lisäys
 - rekursio: unicode, default
 - esim. tabular selitys
 - tekstimuoto
 - kartta?
 - debugger
- refaktorointi --> koodin kommentointi
- makrot? toimiiko ja miten siellä voi olla?
- underscore
- kalvojen numerointi
- tabular-ystö toimimaan lama-monisteen kanssa
- lista tunnistamattomista komennoista
 - erilliseen tekstitiedostoon
 - ruudulle
- syötekohtainen konfiguraatio / override
 - matikka
 - normaali conf
- monimutkaiset operaattorit matikassa
 - summa
 - lim
 - matriisit

Jää tekemättä:

- pseudokoodi
- `do_<macro> = do_<tunnettu komento>`
- lähdeviitenotaation pakottaminen haluttuun tyyliin

Alla asiakkaalta saadut alkuperäiset vaatimukset eriteltyinä sykleittäin prioriteettijärjestyksessä

1:

- section/subsection...
- listat (itemize, enumerate)
- * konfiguraatiotiedosto

2:

- tekstin leveys
- kappalejako: sisennys vai ei

- sisällysluettelo
- label/ref

- kuvat
- taulukot
- lähdeviitteet

3:

- matematiikka
- oma conf tiedosto

Tarkemmat vaatimukset

* yleisiä huomioita

ei tavutusta

riville tulevien merkkien maksimi määrittely mahdollista

sisennystyyli

- joko ei sisennystä ja väli tai
- sisennys ja ei väliä
- ts. noudatetaan kirjoittajan käytäntöä

*dokumentin alku, esim:

```
\documentclass[a4paper,10pt]{article}
\usepackage[finnish]{babel}
\usepackage{graphicx}
\graphicspath{{kuvat/}}
\setlength{\parskip}{1ex}
```

oletusarvoisesti pois

*nimikesivu

```
\author{Harri Laine ja Matti Luukkainen}
\title{Ohjelmistojen mallintaminen}
```

title: Ohjelmistojen mallintaminen
author: Harri Laine ja Matti Luukkainen

*sivurajojen merkintä

käyttäjän määrittelemä merkkijono + sivunumero
oletusarvo page NNN

*sisällysluettelo

1 johdanto 1
2 uml 2
2.1 luokkakaavio 4
2.2 xxx 7

(eli sivunumero heti perään)

*jako tiedostoihin

ei noudateta latex-sorsan tiedostojakoa

```
\include{osa1}
```

määritellään itse, esim. tiedosto/luku

* luvut ja aliluvut

```
\section{Johdantoa ohjelmistotuotantoon}
```

```
\subsection{Vaativuusmäärittely}
```

1. Johdantoa ...

1.1 Vaativuusmäärittely

* listat

```
\begin{itemize}
```

```
\item opetushallinto voi syöttää kurssin tiedot järjestelmään
```

```
\item ...
```

```
\end{itemize}
```

- opetushallinto

- teksti alkaa

teksti jatkuu

konfiguroitava listamerkki

* sisäkkäinen lista

```
\begin{itemize}
```

```
\item opetushallinto voi syöttää kurssin tiedot järjestelmään
```

```
\begin{itemize}
```

```
\item sisäkkäistä asiaa
```

```
\item ...
```

```
\end{itemize}
```

```
\item ...
```

`\end{itemize}`

- opetushallinto
- sisäkkäistä asiaa
- ...
- lisää tekstiä

* numeroidut listat

`\begin{enumerate}`

vastaavatsti:

1. aaa
2. bbb

*quote

tekstiä

`\begin{quote}`

sisennettyä tekstiä

`\end{quote}`

tekstiä

sisennettyä tekstiä

* sisennys

konfiguroitavissa

onko sisennys aina sama? (itemize, sisällysluettelo)

*muotoiltu teksti

`{\em tekstiä} \emph{tekstiä} \textem{tekstiä}`

muotoon `{\em tekstia}`

`{\bf teksti}`

* tavutusohjeet

ta\ -vu\ -te\ -taan

poistetaan

* ignoreitavia komentoja

\mbox
\newpage
\pagebreak
\clearpage
\nepagebreak
\setlength
\flushright
\flushleft
\center
\large, \tiny ym tekstin kokoon vaikuttavat

* suorat muunnokset

\ldots ...
\n (rivinvaihto)
\verbatim

* alaviite

teksti \footnote{}

laitetaan sivun loppuun, ilmaistaan tekstissä kohta johon fn tulee, että alla on footnote

teksti {fn1}

* taulukko eli tabular

```
\begin{tabular}{l c r}  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{tabular}
```

```
1 2 3  
4 5 6  
7 8 9
```

```
\begin{tabular}{|l|c|r|}  
\hline  
111 & 2 & 3333 \\  
\hline  
4 & 555 & 6 \\  
7 & 8 & 9 \\  
\hline  
\end{tabular}
```

111 | 2 | 3333

4 | 555 | 6

7 | 8 | 9

tai kaikki viivat

multicolumn huomioitava luonnollisella tavalla

\arraystretch

\tabcolsep

ym voidaan ignoroida

* kuvat

ilmaistaan, että kuva {figure 10 kuva.eps}

ilmaistaan tiedoston nimi

kuvateksti säilytetään

* kuva, sivu ja kappaleviitteet

\label, \ref, \pageref

ekspandoidaan

*asciimath symbolit:

Tarkennettuja vaatimuksia matematiikan osalta

syntaksi: <http://www1.chapman.edu/~jipsen/mathml/asciimathsyntax.html>

* sulut

käytetäänkö aina samoja sulkuja operaattoreiden parametreissa vai vaihtelee tilanteen mukaan

* välien poisto

$(a+1)(a+2) \rightarrow (a+1)(a+2)$

$(a+1) \rightarrow (a+1)$

$(a+1) \rightarrow (a+1)$

$a+2=3 \rightarrow a+2=3$

eli monta väliä $\rightarrow$ 1 väli

*symbol

latex	hlab	asciimath
<code>\forall</code>	<code>\forall</code>	<code>\forall</code>
<code>\exists</code>	<code>\exists</code>	<code>\exists</code>
<code>\lor</code> <code>\vee</code>	<code>\vee</code>	<code>\vee</code> or
<code>\land</code> <code>\wedge</code>	<code>\wedge</code>	<code>\wedge</code> and
<code>\in</code>	<code>\in</code>	<code>\in</code> (in)
<code>\geq</code>	<code>\geq</code>	<code>\geq</code>
<code>\emptyset</code>	<code>\emptyset</code>	<code>\emptyset</code>
<code>2^{n+1}</code>	<code>2^{n+1}</code>	<code>2^{n+1}</code> tai <code>()</code>
<code>\cup</code>	<code>\cup</code>	<code>\cup</code> union
<code>\cap</code>	<code>\cap</code>	<code>\cap</code> intersets
<code>\setminus</code>	<code>\setminus</code>	<code>\setminus</code> minus
<code>\subset</code>	<code>\subset</code>	<code>\subset</code>
<code>\subseteq</code>	<code>\subseteq</code>	<code>\subseteq</code>
<code>\supset</code>	<code>\supset</code>	<code>\supset</code>
<code>\times</code>	<code>\times</code>	<code>\times</code>
<code>\equiv</code>	<code>\equiv</code>	<code>\equiv</code>
<code>\neq</code>	<code>\neq</code>	<code>\neq</code>

muunneltu fontti kirjaimissa:

`\mathbb{R}` `\mathbb{R}`

`\mathbb{N}` `\mathbb{N}`

muut vastaavasti

kreikkalaiset aakkoset:

`\lambda` `\lambda`

`\alpha` `\alpha`

`\mathbf` käsitellään kuten normaalissa tekstissä

*nuolet

luonnollisella tavalla

`\leftarrow` `\leftarrow`

`\longleftarrow` `\longleftarrow`

`\Leftrightarrow` `\Leftrightarrow` eli aina tuplana jotta ei sotkeudu vertailuihin

`\Leftrightarrow` `\Leftrightarrow`

harpoonit ja hookarrowit kuten nuolet

*hatut

`\hat{XY}` `\hat{XY}`

`\widehat{XY}` `\widehat{XY}`

`\bar{a}` ei vielä päätetty

$\vec{a}$ $\langle a \rangle$
 $\vec{AB}$ $\langle AB \rangle$
 $\overrightarrow{AB}$ $\langle AB \rangle$
 entä muut hattusymbolit?

*parametrilliset operaattorit

$\lim_{n \rightarrow 10}$ $\lim_{n \rightarrow 10}$
 $\sum_{i=1}^n$ $\sum_{i=1}^n$
 vastaavasti joukko-operaatioille
 $\prod \Rightarrow \text{prod}$
 $\frac{n-1}{20}$ $(n-1)/20$ (sulut tarvittaessa)
 $\dfrac{\quad}{\quad}$ samoin kuin edellinen
 p_{ij} p_{ij}
 $p_{\{i,j\}}$ $p_{\{i,j\}}$ tai voi muuttaa ylläolevaksi
 p^3_{ij} sama
 $\sqrt{x+1}$ $\sqrt{x+1}$
 $\sqrt[3]{4}$ $\sqrt[3]{4}$
 $a \cdot b$ $a * b$
 $1 \ldots n$ $1 \dots n$
 myös $\vdots$ ja $\ddots$ merkitään ...
 $a \bmod b$ $a \bmod b$
 $x \equiv a \pmod{b}$ $x == a \pmod{b}$
 $\int_0^{10}$ $\text{int}(0,10)$

*monimutkaiset operaattorit

$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$
 $\text{bin}(n \ k) = \text{b}((n-1) \ k) + \text{b}((n-1) \ (k-1))$

$a \stackrel{\text{def}}{=} 10+x$
 $a \stackrel{\text{def}}{=} 10+x$

$\sum_{\substack{0 \leq i < n \\ j \subseteq i}} P(i,j) = Q(i,j)$

$\sum_{0 \leq i < n; j \subseteq i} i \rightarrow n$

operaattoria tulee edeltää joko välilyönti, rivinvaihto tai sulku

* sulut

$\Big((x+1)(x-1)\Big)^2$
 $((x+1)(x-1))^2$

kaikki sulut normaalikokoisia

*ympäristöt

$E = mc^2$

poistataanko dollarit?

- jos omalla rivillään niin kyllä
- entä jos tekstin seassa: Puun T solmuissa ...
konfiguroitavissa

```
\begin{equation}  
E = mc^2  
\end{equation}
```

kaava (1.1):

$E = mc^2$
jatkuu

ei keskitystä

```
\begin{equation*}  
E = mc^2  
\end{equation*}
```

kaava:

$E = mc^2$

$E=mc^2$ toimii kuten $equation^*$

* suorat muunnokset

$E=mc^2$ \text{Einstein} \quad f=ma \text{Newton}

erilaisia välejä

\quad pieni väli -> space

\quad\quad pitkä väli -> 2*space

\! -> space, myös jos usea \!

* poistetaan

```
\!  
\text{}
```

* jätetään tekstiin

$\underline{\text{teksti}}$
 $\overline{\text{teksti}}$

* align ja eqnarray

käsitellään kuten normaalit taulukot, esim:

```
\begin{align}
a &= b + c \\
&= d + e + j + k + l \nonumber \\
&+ m + n + o \\
&= p + q + r + s \\
\end{align}
```

$$a = b + c \quad (3.10)$$

$$= d + e + j + k + l \\ + m + n + o \quad (3.11)$$

$$= p + q + r + s \quad (3.12)$$

jos määritelty \lefteqn, tehdään muotoilu ilmeisellä tavalla, tyyliin,
esim:

$$\begin{aligned} a+b+c+d+e \\ &= f + g \\ &= q + w \end{aligned}$$

```
\begin{equation*}
|x|=
\begin{cases} x & \& \text{if } x \geq 0, \\
-x & \& \text{if } x \leq 0. \end{cases} \\
\end{equation*}
```

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

jos useampia vaihtoehtoja, niin muoto:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

`\IEEEeqnarray` käsitellään vastaavasti, ks `\tabular`
`\:` voidaan jättää huomioimatta
`\IEEEeqnarraymulticol{}` käsitellään ilmeisellä tavalla

* taulukot ja matriisit

sama asia tehdään joskus seuraavasti:

```
\begin{equation*}
T(k) = \left\{
\begin{array}{ll}
O(1) & \text{kun } k=1 \\
T(k/2)+O(1) & \text{kun } k>1
\end{array}
\right.
\end{equation*}
```

joka renderöitävä myös

$$T(k) = \begin{cases} O(1) & \text{kun } k=1 \\ T(k/2)+O(1) & \end{cases}$$

eli taulukot käsitellään luonnollisella tavalla kuten `\tabular`

`\matrix` ja sen variantit käsitellään luonnollisella tavalla:

```
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

*teoreema, todistus

```
\newtheorem{teor}{Teoreema}
```

```
\begin{teor}
AVL-puun korkeus on  $O(\log n)$ 
\end{teor}
```

jos mahtuu yhdelle riville:

Teoreema 1. AVL-puun korkeus on $O(\log n)$

jos ei niin:

Teoreema 1.

AVL-puun korkeus on $O(\log n)$
ja syvyys myös $O(\log n)$

HUOM: teoreemien numerointi!

```
\begin{proof}
oletetaan että ...
\end{proof}
```

Proof:
oletetaan että ... MOT

* mietinnän alla

```
\underbar{a+b+c}_{summa}
{a+b+c}_{summa}
```

```
\overbrace{vast}
```

```
\multiline ehkä huomioimatta
```

Tarkennettuja vaatimuksia lama-monisteen osalta

Konfigurointia?

```
[environments] // suuren ympäristöt \begin{...}\end{...}
array -> tabular
mycode -> clsrsrcode2e
```

```
[singles] // yksinkertaiset komennot
Blue -> remove
Green -> remove
```

```
[?????] // muut siltä väliltä
emphasis -> {em \1}
em -> emphasis
bem -> emphasis
rem -> emphasis
```

seuraavasta ei kannata välittää:

```
emphasis: {\ []}+ -> {\ \1} (regexp)
em: {\ []}+ -> emphasis
bem: {\[a-z]em []}+ -> emphasis, eli myös esim rem samoin
värit: \Blue, \Green -> remove
```

kivisen makrot: // tiedostokohtainen konf

`\bem` -> `\em`

`\Blue` -> poistetaan

`\begin{mydesc}` -> normaalisti

`\begin{slide}` -> `s1,s2,s3,...`

`\set{...}` -> `{...}`

`\mid` -> `|`

`S \derives v_1` muotoon `S -> v_1`

`S \dderives v_1` muotoon `S => v_1`

`a \leadsto b` -> `a leads b`

`\begin{mycode}` -> "hyvin", eli kuten `\usepackage{clrscode3e}` tai 2e

`\lname{noun-phrase}` -> `<noun-phrase>`

`\tname{hamburger}` -> `hamburger`

`\newcommand{\derTMP}[2]{\proc{Derives}(\#1,\mathrm{\#2})}`

`\derTMP` muotoon `Derives`

muut:

`q_0 \stackrel{\{\Blue 0\}}{\rightarrow} q_1`

muodossa: `q_0 ->\{0\} q_1`

`\esmove` ja `\esmovec` kuten `leadsto+stackrel`

`\overline{A}`

muodossa: `A^c`

`\mathrm{}` kuten `\mathbb`

`\ast` -> `^*`

`\delta` ~d

`\Sigma` ~S

`\varepsilon` ~e

`\Gamma` ~G

`\bigcup\{a \in A\}` -> kuten `\sum`

`\rma,\rmb,\rmc,\rmd` -> `a, b, c, d`

`\underbrace{R * \ldots * R}_{\mbox{$k$ kertaa}}`

muotoon: $\{R * \dots * R\}_{k \text{ kertaa}}$

`\rm` -> pois

`\vdash` merkataan |-

`\blank` merkataan ! (konfiguroitava, dokumenttikohtainen)

`\hash` merkitään #

`\code{xx}` -> <xx>

2.2 Ei-toiminnalliset vaatimukset

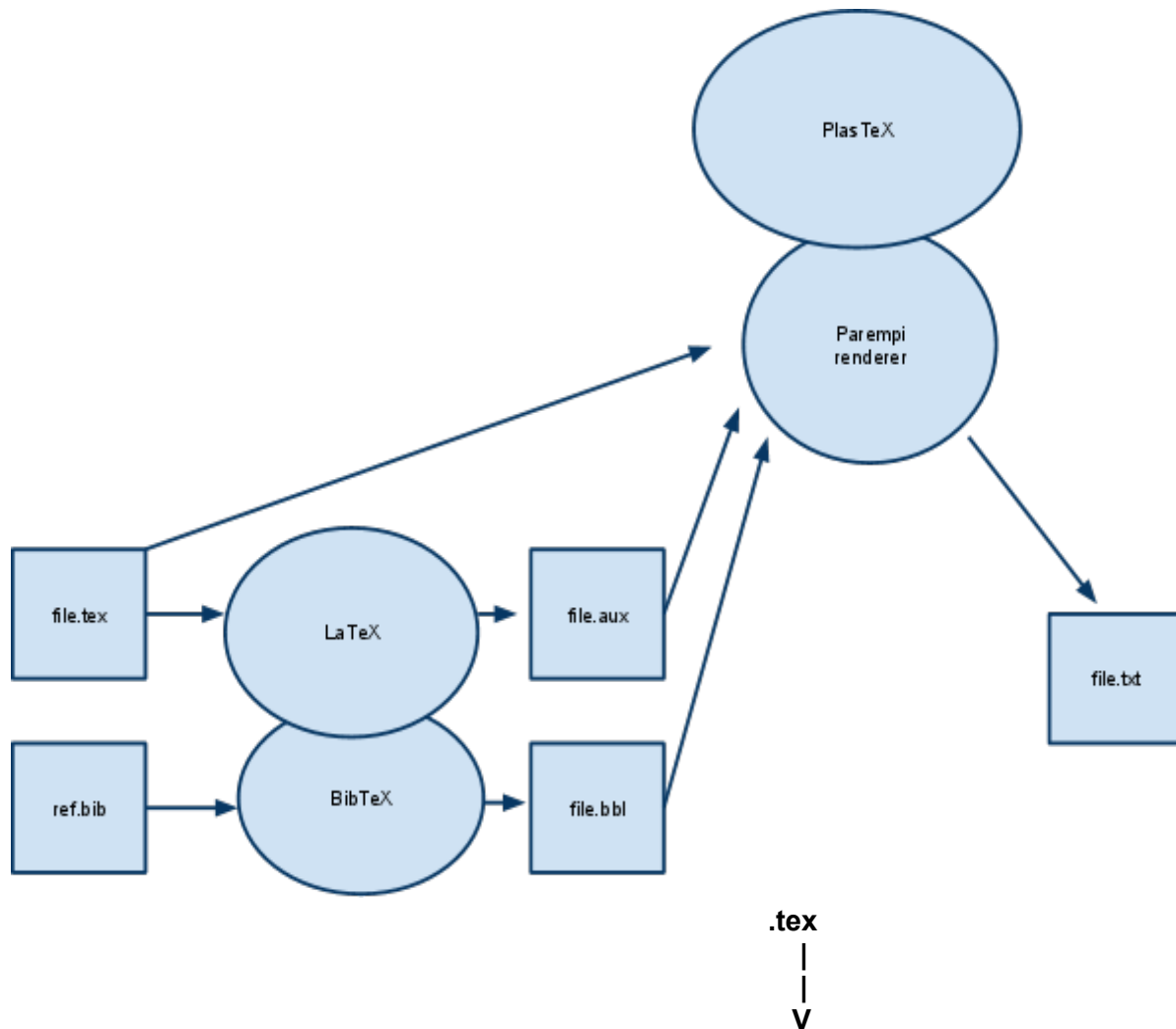
Toteutettavan ohjelman tulee toimia latexia vastaavassa ajassa. Kääntämisaika toteutetulla ohjelmalla ei siis saisi olla moninkertainen verrattuna latex-ohjelmiston kääntämiseen käyttämään aikaan. Ohjelman on myös tuotettava samasta .tex-tiedostosta jokaisella ajokerralla samanlainen .txt-tiedosto yllä määriteltyjen vaatimusten mukaisesti. Lopputulos ei siis saa vaihdella ajokerrasta toiseen.

Ohjelman asennuksen on myös oltava sellainen, että tavallinen tkkl-opiskelija pystyy sen suorittamaan seuraamalla mukana toimitettavia asennusohjeita. Tuotettavan tekstitiedoston formaatin on oltava luettavissa helposti pistenäytöltä. Lisäksi ohjelmaan tutustumiskynnyksen on oltava apuna käytetyn plasTeX-ohjelmiston tutustumiskynnystä alempi. Viimeinen vaatimus on käytännössä toteutettu yrittämällä dokumentoida myös erinäisiä plasTeX:n osia mahdollisimman selkeästi.

3. Rakenne

3.1 Yleisrakenne

- kaaviona



.tex --> Latex --> .aux ja .toc --> bibtex --> .bib --> Plastex --> Rendereri --> .txt

3.2 Osat

- `__init__.py`
 - Renderers kansion alikansiossa parempi sijaitseva toteutettu rendereri
 - Plastex ajettaessa tämä renderöi .tex-tiedoston .txt-tiedostoksi vaatimusten osoittamalla tavalla
 - Otetaan käyttöön .plasTeXrc-tiedostossa määrittelemällä `renderer=parempi`
- `.plasTeXrc`
 - Käyttäjän kotihakemistossa sijaitseva konfigitiedosto
- `.lattertextsymbols`
- `slides.py`

4. Dynaaminen käyttö

4.1 Skenaario 1

- .tex-tiedoston kääntämäinen .txt-tiedostoksi

confien kanssa, sisällysluettelon kanssa, bibtex

- type: system operation or use case
- description of the scenario
- how scenario interacts with components

5 Other Views

5.1 Process: in a running system, how components are divided as processes

--> e.g. There are can be multiple instances of KujeProcessor, but can there be multiple instances of KujeServer?

--> Has multiple instances of KujeProcessor ever been tested? Has multiple instances of KujeServer ever been tested?

5.2 Development: Where in code the codebase components are implemented

5.3 Physical: How components are separated in the hardware level

--> e.g. Division between the frontend and the backend; list also all physical dependencies between components, e.g. "KujeProcessor and WebApp must stay on the same physical server because..."

5.4 Deployment: How components are deployed and possible constraints