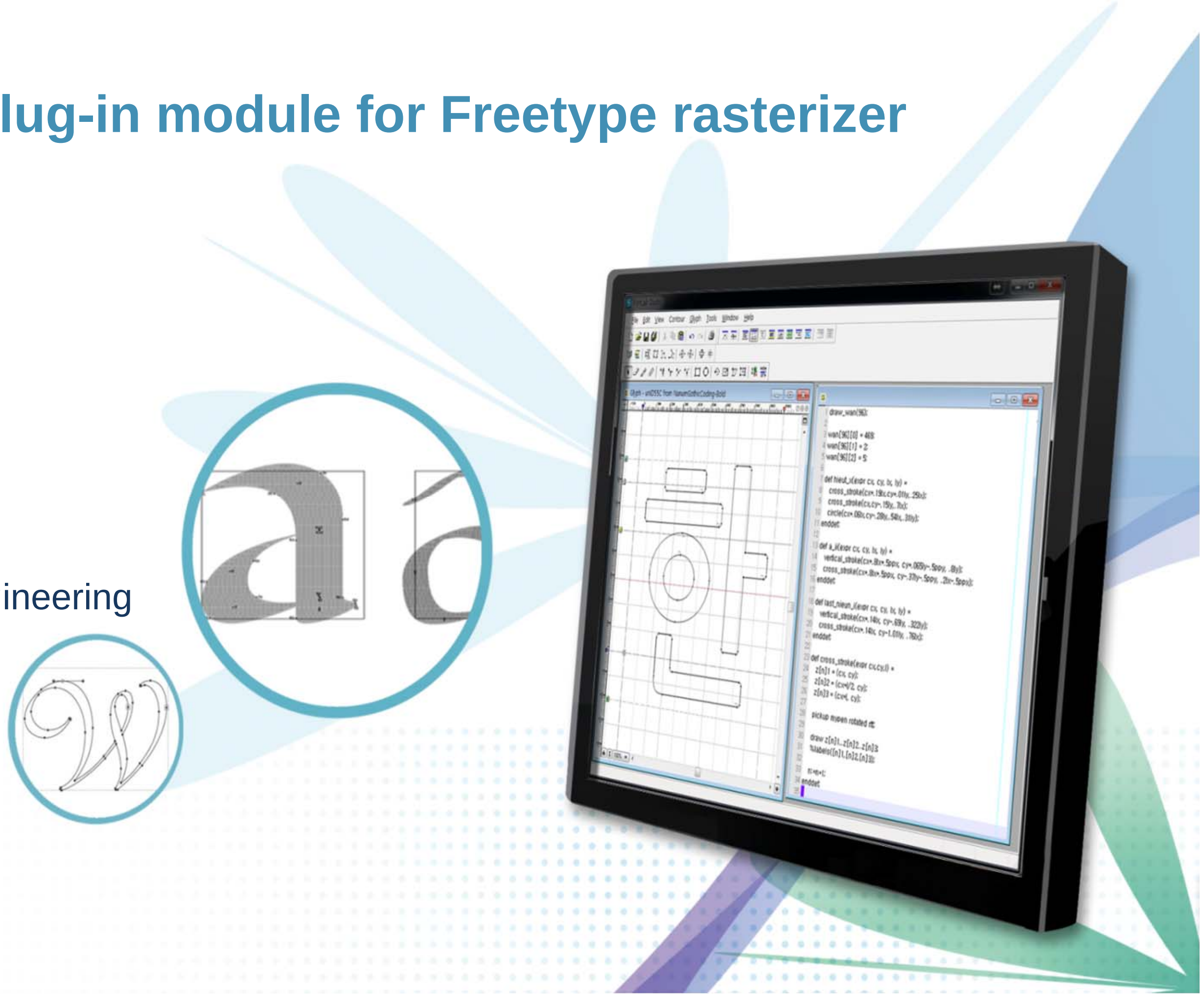
MFCONFIG

- METAFONT plug-in module for Freetype rasterizer



```
draw_wan(96);

wan[96][0] = 469;
wan[96][1] = 2;
wan[96][2] = 5;

def hieut_i(expr cx, cy, lx, ly) =
  cross_stroke(cx+.19lx,cy+.01ly,.25lx);
  cross_stroke(cx,cy-.15ly,.7lx);
  circle(cx+.08lx,cy-.28ly,.54lx,.31ly);
enddef

def a_i(expr cx, cy, lx, ly) =
  vertical_stroke(cx+.8lx+.5pox, cy+.065ly-.5poy, .8ly);
  cross_stroke(cx+.8lx+.5pox, cy-.37ly-.5poy, .2lx-.5pox);
enddef

def last_nieun_i(expr cx, cy, lx, ly) =
  vertical_stroke(cx+.14lx, cy-.69lx, .322ly);
  cross_stroke(cx+.14lx, cy-1.01ly, .76lx);
enddef

def cross_stroke(expr cx,cy,l) =
  z[n]1 = (cx, cy);
  z[n]2 = (cx+l/2, cy);
  z[n]3 = (cx+l, cy);

  pickup mypen rotated rt;

  draw z[n]1..z[n]2..z[n]3;
  %labels([n]1,[n]2,[n]3);

  n:=n+1;
enddef
```

2016. 07. 26.

Jaeyoung Choi

School of Computer Science & Engineering
Soongsil University, Seoul, Korea

숭실대학교
Soongsil University

- Contents -

01 **MOTIVATION**

02 **DESIGN GOALS**

03 **MFCONFIG MODULE**

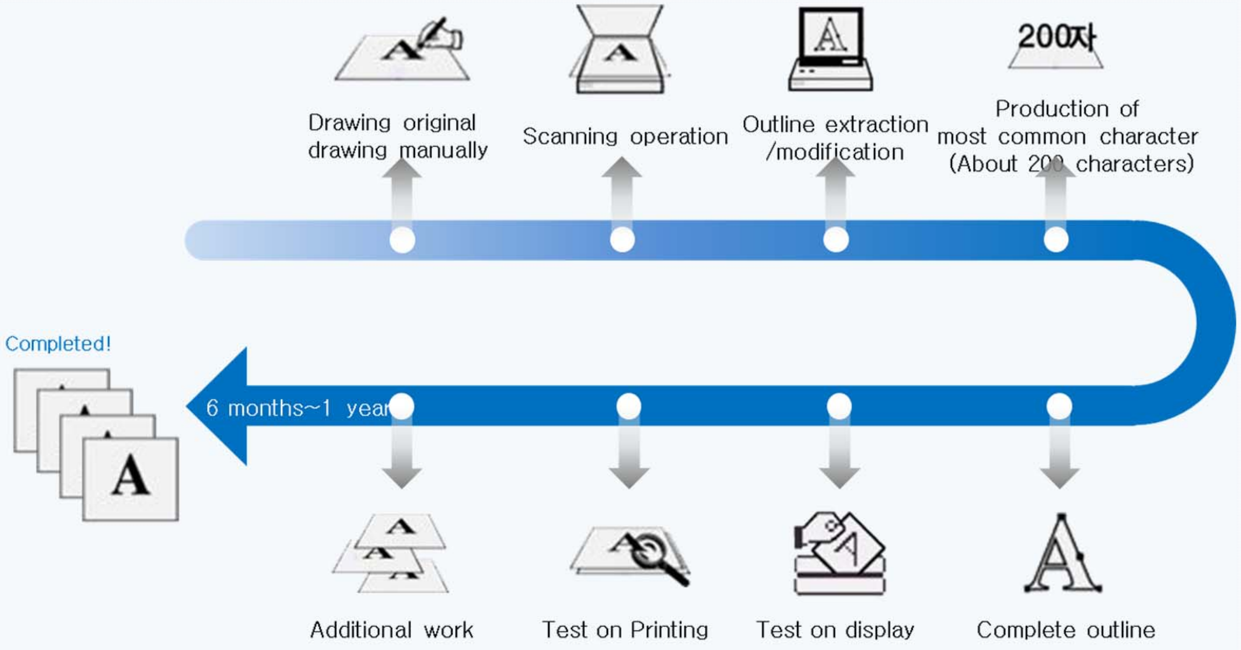
04 **EXPERIMENT**

01 MOTIVATION

- Font Design process -

Font Design Process

- Take much time for developing font file, especially the CJK font
  - Spend much time and cost
  - Repeat same tasks if a different style font required every time


200자
Drawing original drawing manually
Scanning operation
Outline extraction /modification
Production of most common character (About 200 characters)
Completed!
6 months~1 year
Additional work
Test on Printing
Test on display
Complete outline

01 MOTIVATION - Font Design process -

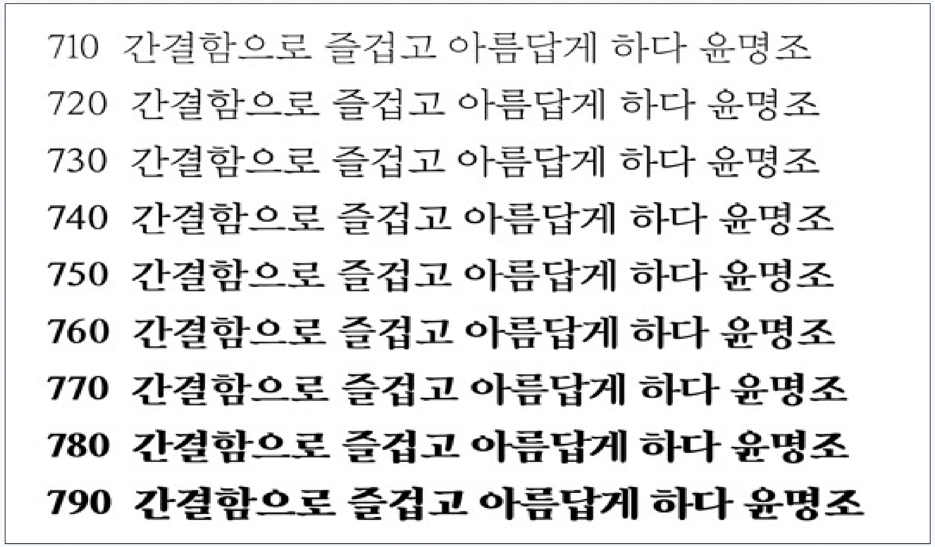
❖ Font Design Process

Yoon-Ming 700 Family

- ❖ Development Period : **4 years (2010~2014)**
- ❖ Manufacturer: Yoon Design Group

  General Design Manager(1), Typeface Designer(4)
- ❖ Consists of a **9 stages(710~790) family**
- ❖ Korean 11,172, English 94, Symbol 986 (Extended Symbol 1,818), Chinese 4,888

가나다라마
바사아자차
카타파하
함함함함함함함함함
와 와 와
와 와 와
와 와 와
뫄 뫼 맥
뫄 뫼 맥
뫄 뫼 맥



710 간결함으로 즐겁고 아름답게 하다 윤명조
720 간결함으로 즐겁고 아름답게 하다 윤명조
730 간결함으로 즐겁고 아름답게 하다 윤명조
740 간결함으로 즐겁고 아름답게 하다 윤명조
750 간결함으로 즐겁고 아름답게 하다 윤명조
760 간결함으로 즐겁고 아름답게 하다 윤명조
770 간결함으로 즐겁고 아름답게 하다 윤명조
780 간결함으로 즐겁고 아름답게 하다 윤명조
790 간결함으로 즐겁고 아름답게 하다 윤명조

01 MOTIVATION - Programmable font -

❖ Programmable (Algorithmic) font

- Consist of programs to derive fonts
- Include not one set of a font, but the font family
- CSD(Character Simulated Design), METAFONT

- Advantage
  - Provide parameters to define font styles
  - Various fonts can be generated by changing the values

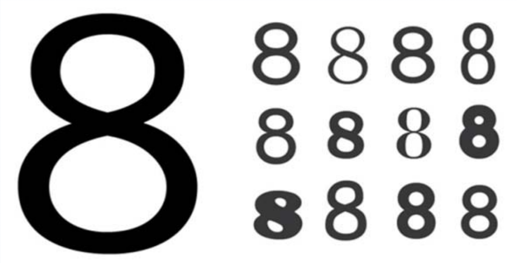
- Disadvantage
  - Require programming skills

01 MOTIVATION - METAFONT -

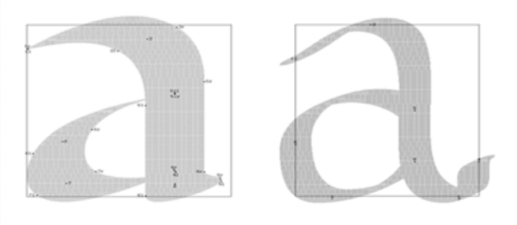
METAFONT

- Developed by Donald Knuth at Stanford University [1986]
- Programming language to create fonts based on a basic skeleton
- Skeleton method rather than outline method
- Pen's shape and paths are important factors to determine the style of the character
- Have great expressive power for font expression
- But not easy to use for font designers, who are not familiar with programming languages

- Advantage
  - Easy to generate letters with designing a pen and a trace
  - Various fonts can be generated easily
  - Small storage capacity

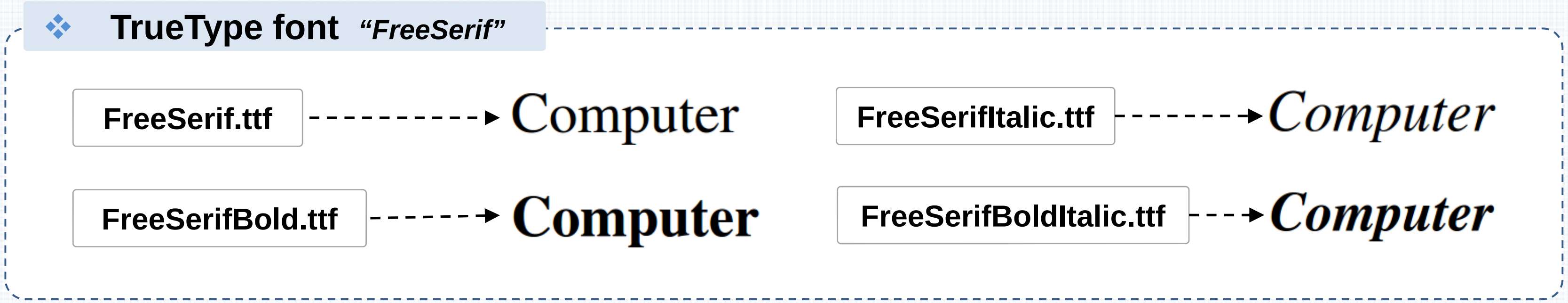


- Disadvantage
  - Not easy for programming
  - No visual editor
  - Need computing power to generate fonts



01 MOTIVATION - METAFONT -

Various Style change of METAFONT vs TrueType font



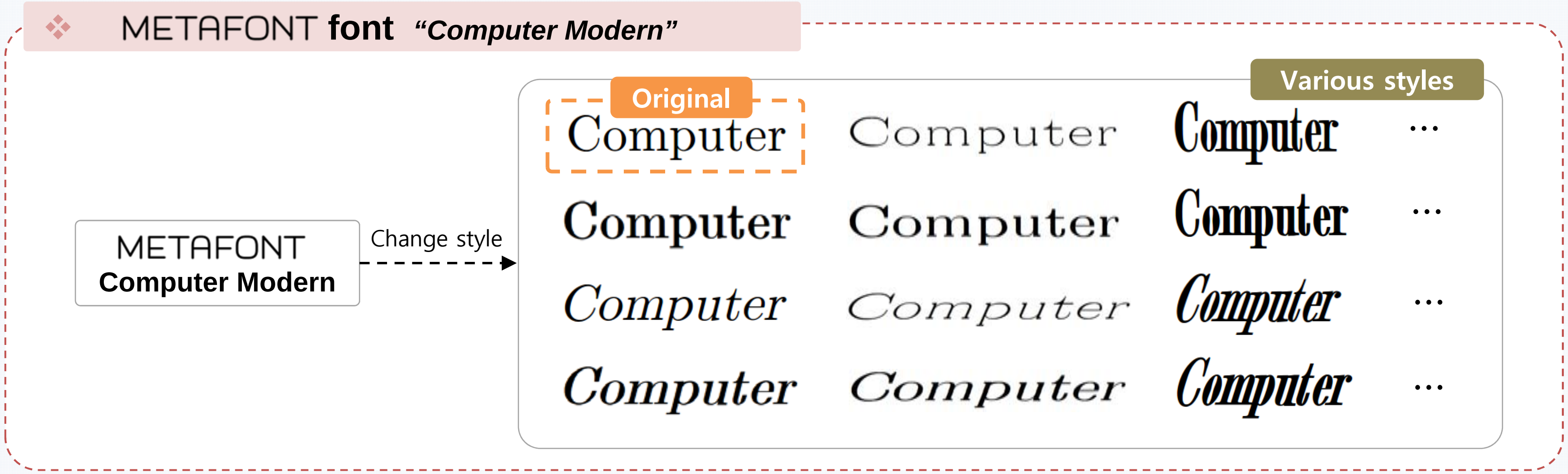
TrueType font *“FreeSerif”*

FreeSerif.ttf → Computer

FreeSerifItalic.ttf → *Computer*

FreeSerifBold.ttf → **Computer**

FreeSerifBoldItalic.ttf → ***Computer***



METAFONT font *“Computer Modern”*

METAFONT
Computer Modern

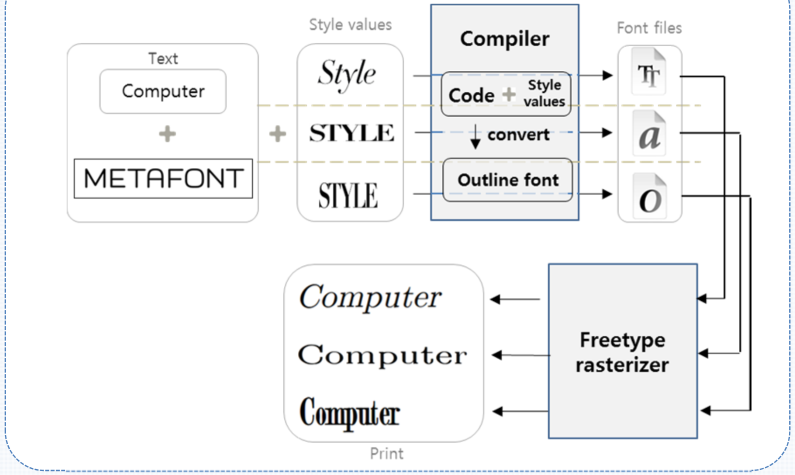
Change style
Original
Various styles
Computer Computer Computer …
Computer Computer Computer …
Computer Computer Computer …
Computer Computer Computer …

01 MOTIVATION

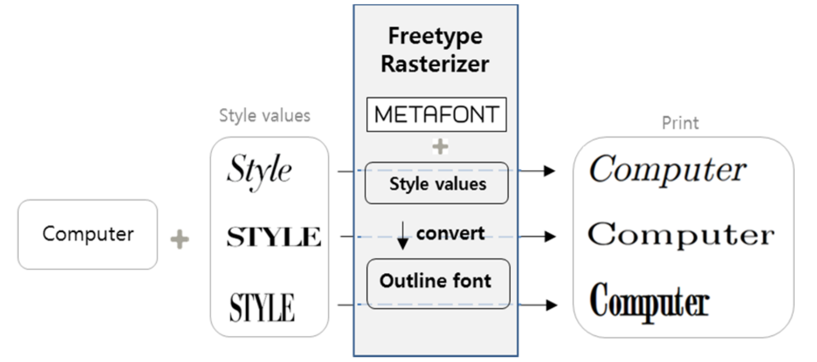
Use METAFONT in Freetype rasterizer directly

- Don't need to pre-convert METAFONT to outline font for Freetype rasterizer
  - Use "mf" file like Truetype font(TTF) files
- Just use METAFONT like general digital fonts
  - Ensure High Performance of PC environment
- Change font style freely what user want
  - Available changing font style by style values easily

Pre-converting work



Use it directly !!



02 DESIGN GOALS - main idea -

4 Main concept of *MFCONFIG* Module

Easy installation

- Making a module which can be installing to current font system naturally
- METAFONT files can be setup easily

Easy to use METAFONT

- User and application can use METAFONT without any pre-working

ex) convert METAFONT to other font file

Easy to change font styles

- Input font style what user want

ex) slant, stroke, skeleton, etc

Style STYLE STYLE

High Portability

- Making a module which is available to Freetype rasterizer
- Consider which related libraries with Freetype rasterizer

02 DESIGN GOALS - architecture -

Combine 2 features

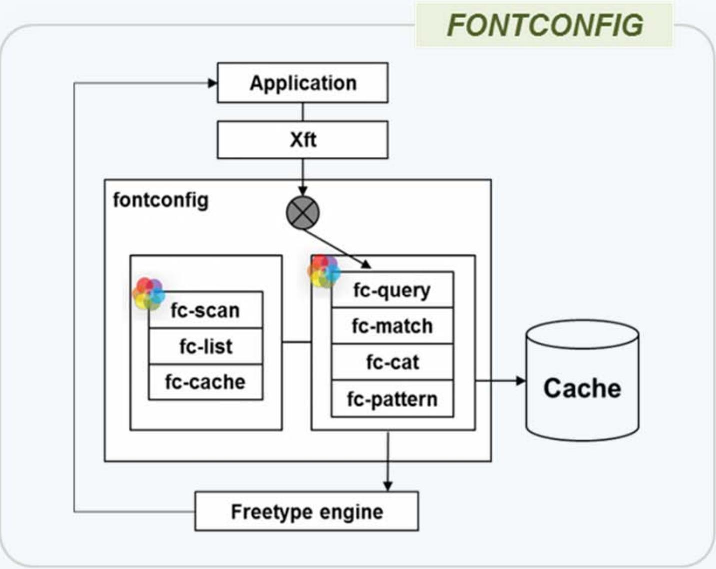
- The process for the printing of digital fonts in “FONTCONFIG” lib
- Consider inner commands of FONTCONFIG (FONTCONFIG ships)

< K. Packard. Fontconfig. Gnome User's and Developers European, 2002 >

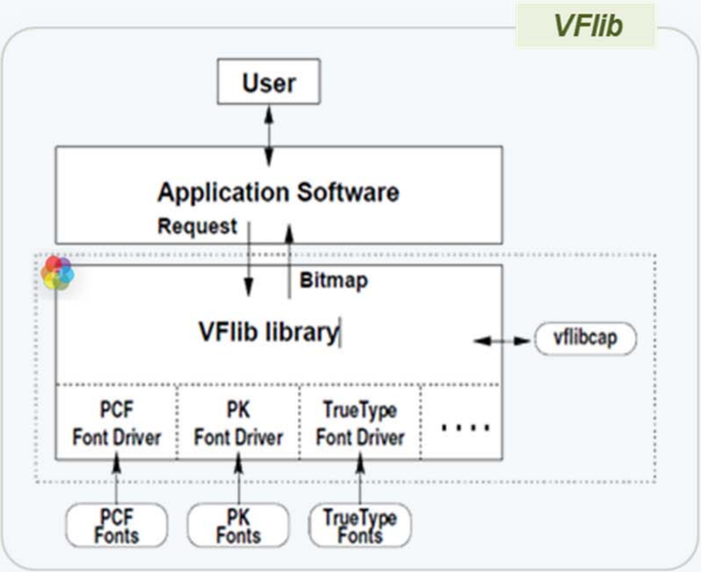
+

- The font driver architecture of “VFlib” (Virtual font library)

< H.Kakugawa. A general purpose font module for multilingual application programs. SP&E, March 2000 >



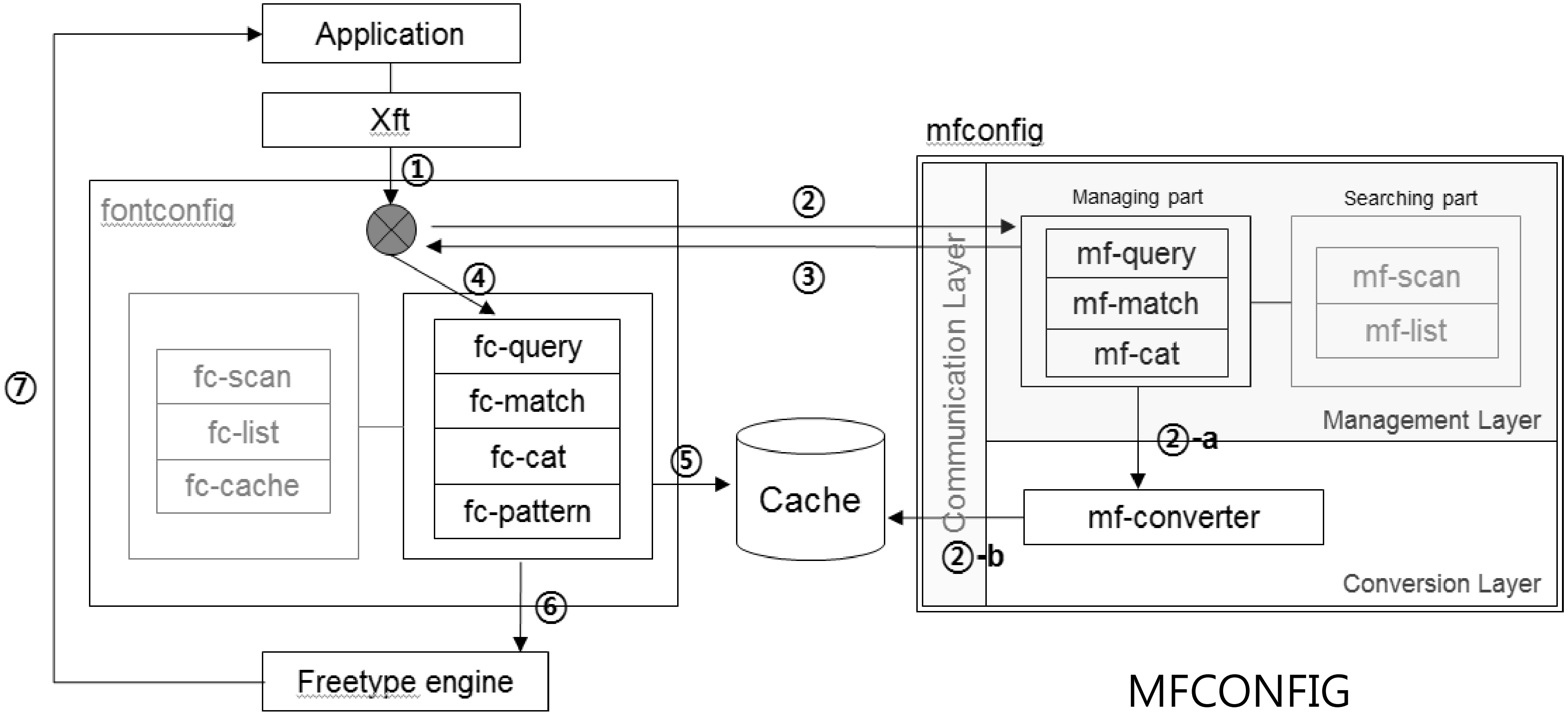
+



03

MFCONFIG : METAFONT plug-in module for Freetype rasterizer

MFCONFIG MODULE



MFCONFIG

FONTCONFIG

03 MFCONFIG MODULE

3 Main Components

Communication

- Interface between *FONTCONFIG* and *MFCONFIG*
- The main functions
  (1) Deliver the request of METAFONT information to the 'Management' unit
  (2) Return the results to *FONTCONFIG*
  (3) Store the outline font file in cache memory

Management

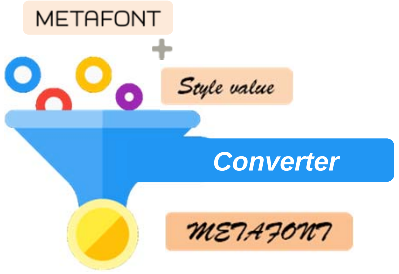
Searching

- Find all of the installed METAFONT fonts
- Store the information in a list

Managing

- Core process of the *MFCONFIG* module

Conversion



- Convert the METAFONT font into the matched outline font
  (If the font is not stored in the cache)
- Store the resulting outline font in the cache
- Notify a message of 'the font is ready' to *FONTCONFIG*

03 MFCONFIG MODULE

Management Component

- The major programs of the MFCONFIG module
- In charge of “***Searching***” and “***Managing***”

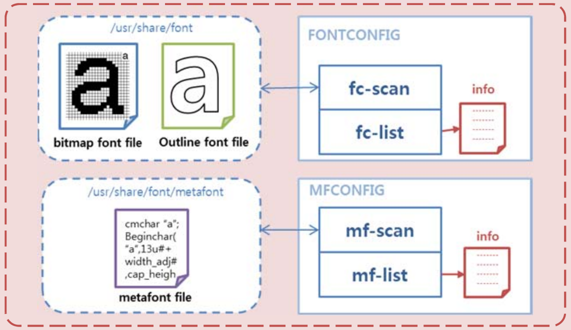
Searching

- Independent function of finding all of the installed METAFONT fonts and storing the information in a list

  (Checking whether a specific font is installed or not and for fetching its information quickly)

- Implemented with *mf-scan* and *mf-list*

  (work similarly to *fc-scan* and *fc-list* in *FONTCONFIG*)



Managing

- Core process of the MFCONFIG module
- Actions:

  (1) Checking if the requested font is prepared in the list

  (2) Checking if the corresponding outline font with styles is stored in the cache memory

  (3) If not in cache memory, send a request to the converter to convert the METAFONT font into the corresponding outline font

- (If the outline font has already been in the cache) a notification is sent directly to *FONTCONFIG*

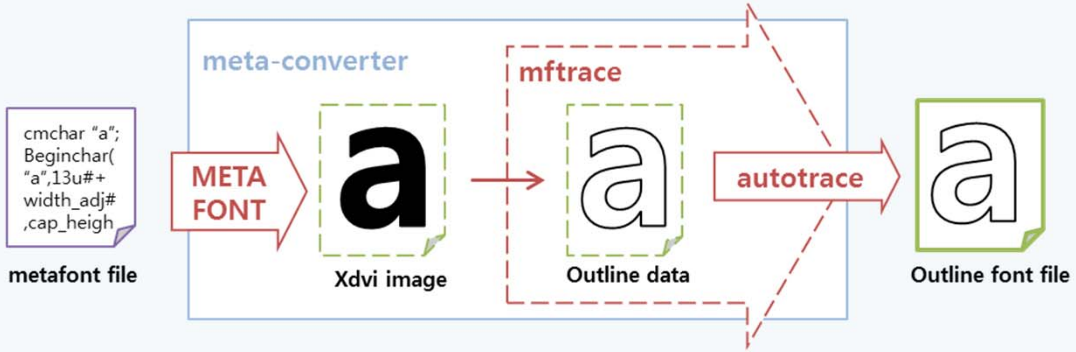
03 MFCONFIG MODULE

Conversion Component

- If the font is not stored in the cache,
  then the Converter converts the METAFONT font
  into the corresponding outline font by applying the style parameters

- *MFCONFIG* module uses *“mftrace”* and *“autotrace”*

- The resulting outline font is then stored in the cache,
  and a notification commands *FONTCONFIG* to use the font.

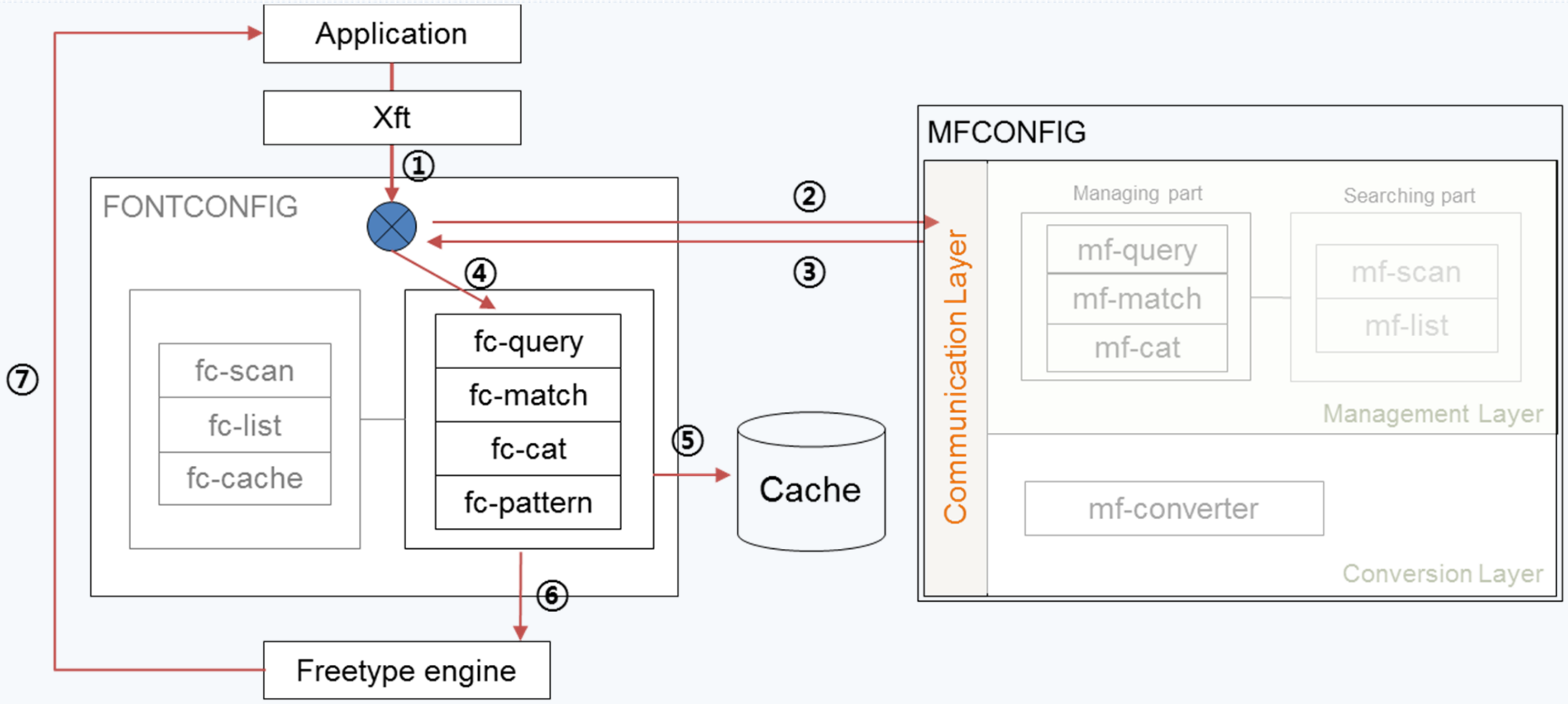


03 MFCONFIG : METAFONT plug-in module for Freetype rasterizer

MFCONFIG MODULE

Scenario 1

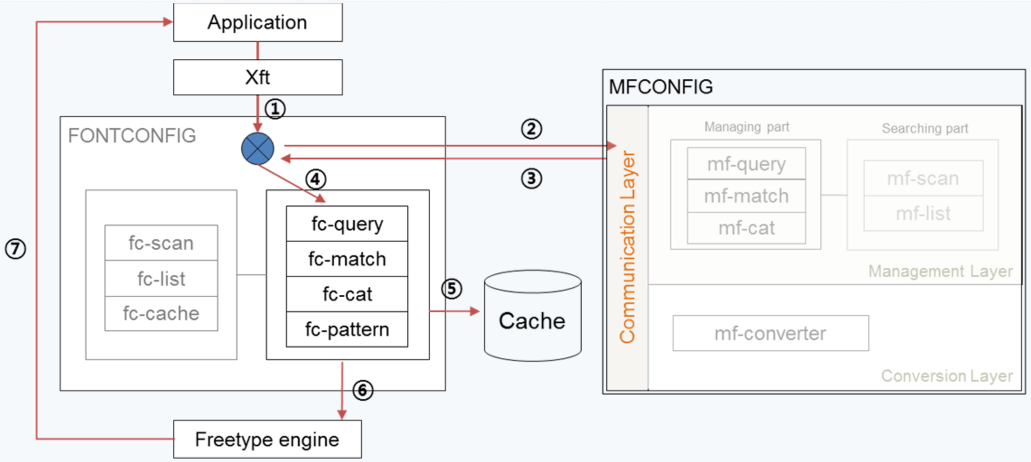
- Requested font is not METAFONT



MFCONFIG MODULE - Scenario 1 -

Scenario 1

- Requested font is not METAFONT



- Running sequence

① Request font file with style value

② Send font information from *FONCONFIG* to *MFCONFIG*
Communication Component checking that requested font is not METAFONT

③-④ Received answer from *MFCONFIG*, use default system font

⑤ Get prepared font data from font cache

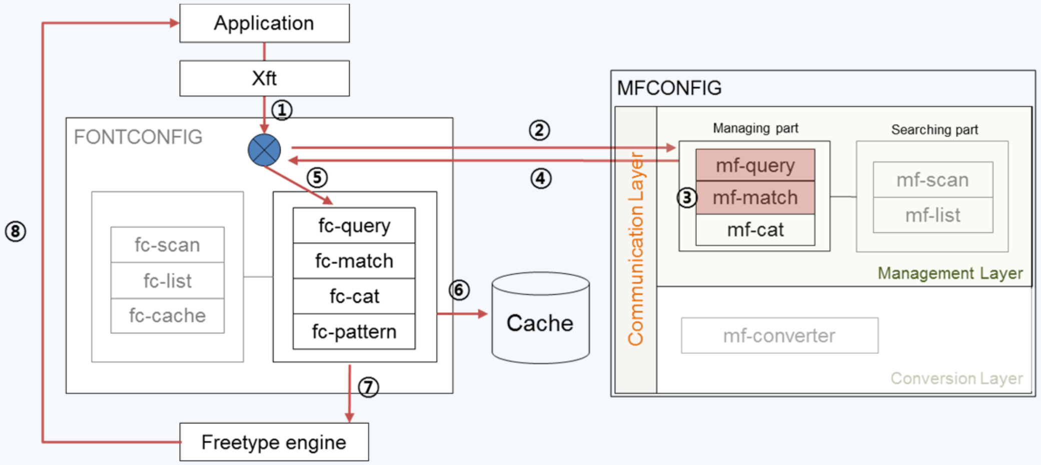
⑥ Send font data to Freetype rasterizer

⑦ Print font screen (default system font)

03 MFCONFIG MODULE - Scenario 2 -

Scenario 2

- Requested font is METAFONT, but this font is not installed in this system



- Running sequence

① Request font file with style value

② Send font information from *FONCONFIG* to *MFCONFIG*
Communication Component checking that requested font is METAFONT

③ *(mf-match)* Try to find requested font in mf-list, but it is not in list

④ Received answer from *MFCONFIG*, use default system font

⑤ Get prepared font data from font cache

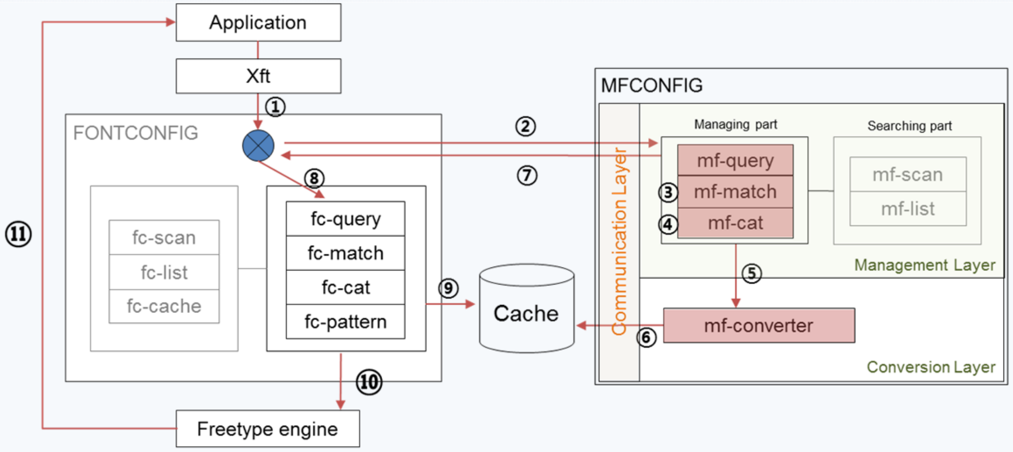
⑥ Send font data to Freetype rasterizer

⑦ Print font screen (default system font)

03 MFCONFIG MODULE - Scenario 3 -

Scenario 3

- Requested font is METAFONT, already installed, but it isn't in the font cache



- Running sequence

① Request font file with style value

② Send font information from *FONCONFIG* to *MFCONFIG*
Communication Component checking that requested font is METAFONT

③ (mf-match) Found requested font in mf-list

④ (mf-cat) Try to find converted outline font in font cache, but it is not prepared

⑤- ⑥ (mf-converter) Convert METAFONT to corresponding outline font and Store it to the cache

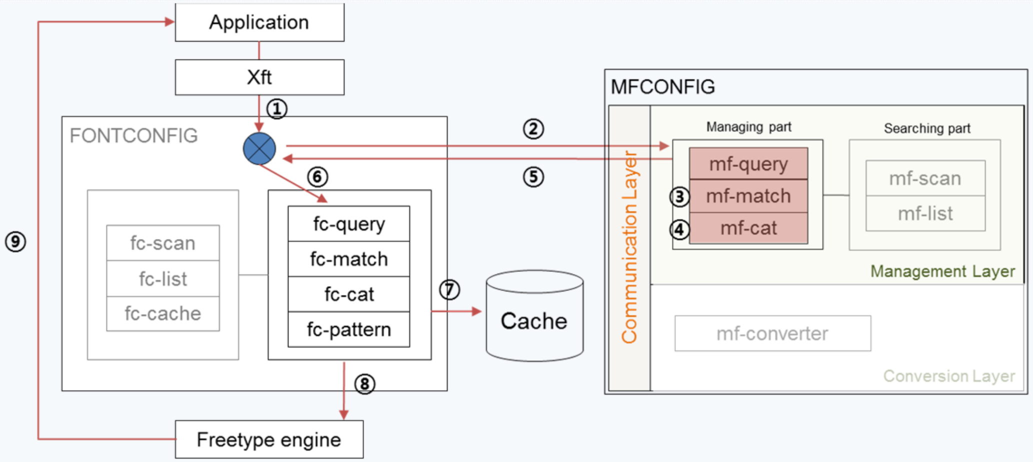
⑦ Send answer to *FONTCONFIG* in order to use converted outline font

⑧,⑨,⑩,⑪ Received answer from *MFCONFIG*, use this converted outline font from the cache

03 MFCONFIG MODULE - Scenario 4 -

Scenario 4

- Requested font is METAFONT, already installed, also it is stored in the font cache



- Running sequence

①　Request font file with style value

②　Send font information from *FONCONFIG* to *MFCONFIG*

Communication Component checking that requested font is METAFONT

③　(mf-match) Found requested font in mf-list

④　(mf-cat) Found converted outline font in font cache

⑤　Send answer to *FONTCONFIG* in order to use converted outline font

⑥,⑦,⑧,⑨　Received answer from *MFCONFIG*, use this converted outline font from the cache

04 EXPERIMENT *MFCONFIG* MODULE

- **Use an application for the use of the X Window system in Linux**

- **Display a text file using two different font files**
  - The TrueType font family named *“FreeSerif“*
    (*FreeSerif.ttf, FreeSerifItalic.ttf, FreeSerifBod.ttf* and *FreeSerifBoldItalic.ttf*)
  - METAFONT font named *“Computer Modern"*
  - The sample text comprises over 2,000 words and over 8,800 characters

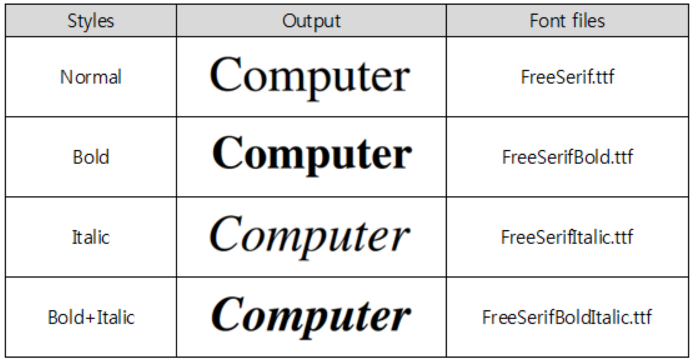
04 EXPERIMENT *MFCONFIG* MODULE

❖ Test font files “FreeSerif” (TTF) , “Computer Modern” (METAFONT)

❖ ***“FreeSerif”*** **font family with the four different styles**

Normal, Bold, Italic, Bold+Italic

(FreeSerif.ttf, FreeSerifBold.ttf, FreeSerifItalic.ttf, FreeSerifBoldItalic.ttf)

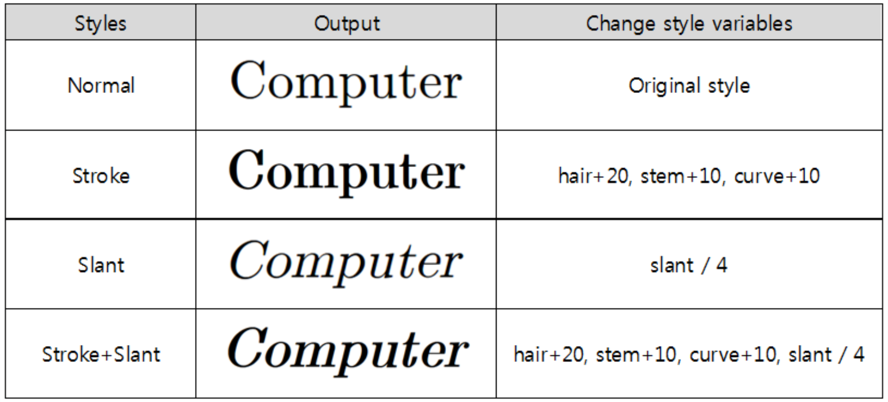


Styles	Output	Font files
Normal	Computer	FreeSerif.ttf
Bold	**Computer**	FreeSerifBold.ttf
Italic	*Computer*	FreeSerifItalic.ttf
Bold+Italic	***Computer***	FreeSerifBoldItalic.ttf

❖ **12 styles that were generated for the *“Computer Modern”* font of** METAFONT

Normal, Stroke, Slant, Stroke+Slant

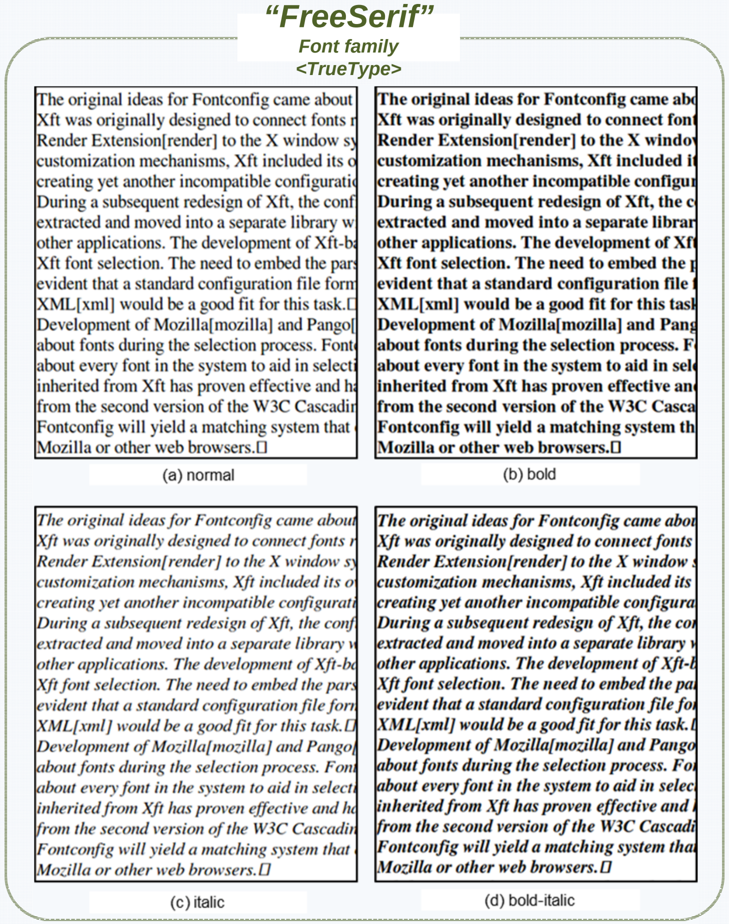
❖ **All of these styles were made from one original prototype of the** METAFONT **font by simple changing of the style parameters*.***



Styles	Output	Change style variables
Normal	Computer	Original style
Stroke	**Computer**	hair+20, stem+10, curve+10
Slant	*Computer*	slant / 4
Stroke+Slant	***Computer***	hair+20, stem+10, curve+10, slant / 4

04 EXPERIMENT *MFCONFIG* MODULE

Test for printing text



“FreeSerif”
Font family
<TrueType>

The original ideas for Fontconfig came about
Xft was originally designed to connect fonts r
Render Extension[render] to the X window sy
customization mechanisms, Xft included its o
creating yet another incompatible configuratio
During a subsequent redesign of Xft, the conf
extracted and moved into a separate library w
other applications. The development of Xft-ba
Xft font selection. The need to embed the pars
evident that a standard configuration file form
XML[xml] would be a good fit for this task.□
Development of Mozilla[mozilla] and Pango[
about fonts during the selection process. Font
about every font in the system to aid in selecti
inherited from Xft has proven effective and ha
from the second version of the W3C Cascadin
Fontconfig will yield a matching system that
Mozilla or other web browsers.□

(a) normal

**The original ideas for Fontconfig came abo
Xft was originally designed to connect font
Render Extension[render] to the X windov
customization mechanisms, Xft included it
creating yet another incompatible configur
During a subsequent redesign of Xft, the c
extracted and moved into a separate librar
other applications. The development of Xft
Xft font selection. The need to embed the p
evident that a standard configuration file f
XML[xml] would be a good fit for this task
Development of Mozilla[mozilla] and Pang
about fonts during the selection process. F
about every font in the system to aid in sele
inherited from Xft has proven effective an
from the second version of the W3C Casca
Fontconfig will yield a matching system th
Mozilla or other web browsers.□**

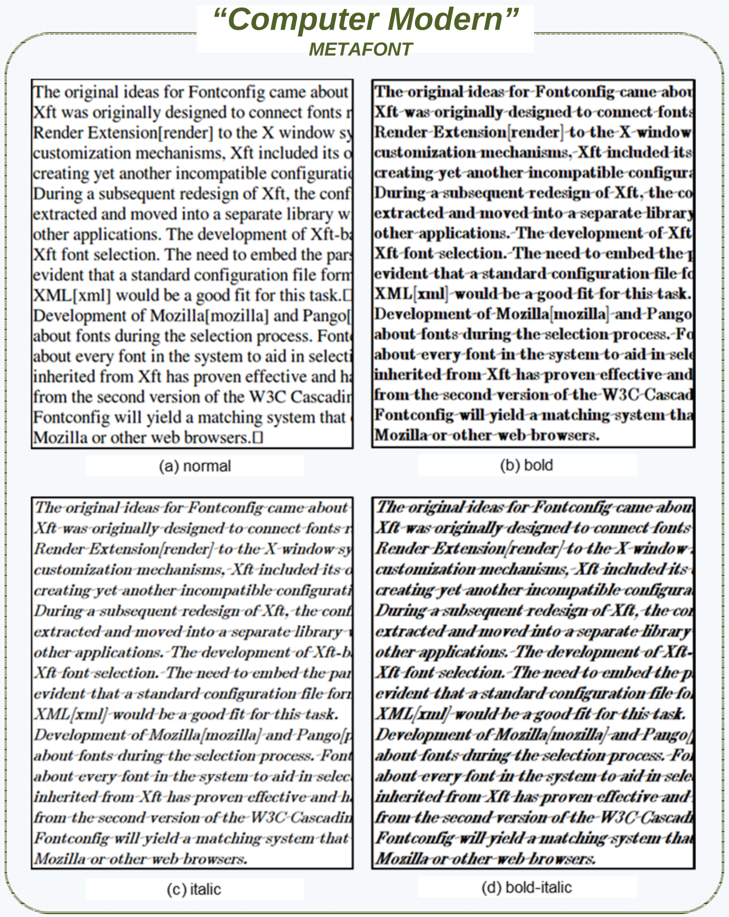
(b) bold

*The original ideas for Fontconfig came about
Xft was originally designed to connect fonts r
Render Extension[render] to the X window sy
customization mechanisms, Xft included its o
creating yet another incompatible configurati
During a subsequent redesign of Xft, the conf
extracted and moved into a separate library w
other applications. The development of Xft-ba
Xft font selection. The need to embed the pars
evident that a standard configuration file form
XML[xml] would be a good fit for this task.□
Development of Mozilla[mozilla] and Pango[
about fonts during the selection process. Font
about every font in the system to aid in selecti
inherited from Xft has proven effective and ha
from the second version of the W3C Cascadin
Fontconfig will yield a matching system that
Mozilla or other web browsers.□*

(c) italic

***The original ideas for Fontconfig came abou
Xft was originally designed to connect fonts
Render Extension[render] to the X window
customization mechanisms, Xft included its
creating yet another incompatible configura
During a subsequent redesign of Xft, the co
extracted and moved into a separate library
other applications. The development of Xft-
Xft font selection. The need to embed the pa
evident that a standard configuration file fo
XML[xml] would be a good fit for this task.
Development of Mozilla[mozilla] and Pango
about fonts during the selection process. Fo
about every font in the system to aid in selec
inherited from Xft has proven effective and
from the second version of the W3C Cascadi
Fontconfig will yield a matching system that
Mozilla or other web browsers.□***

(d) bold-italic



“Computer Modern”
METAFONT

The original ideas for Fontconfig came about
Xft was originally designed to connect fonts r
Render Extension[render] to the X window sy
customization mechanisms, Xft included its o
creating yet another incompatible configurati
During a subsequent redesign of Xft, the conf
extracted and moved into a separate library w
other applications. The development of Xft-ba
Xft font selection. The need to embed the pars
evident that a standard configuration file form
XML[xml] would be a good fit for this task.□
Development of Mozilla[mozilla] and Pango[
about fonts during the selection process. Font
about every font in the system to aid in selecti
inherited from Xft has proven effective and ha
from the second version of the W3C Cascadin
Fontconfig will yield a matching system that
Mozilla or other web browsers.□

(a) normal

**The original ideas for Fontconfig came abou
Xft was originally designed to connect fonts
Render Extension[render] to the X window
customization mechanisms, Xft included its
creating yet another incompatible configura
During a subsequent redesign of Xft, the co
extracted and moved into a separate library
other applications. The development of Xft
Xft font selection. The need to embed the p
evident that a standard configuration file fo
XML[xml] would be a good fit for this task.
Development of Mozilla[mozilla] and Pango
about fonts during the selection process. Fo
about every font in the system to aid in sele
inherited from Xft has proven effective and
from the second version of the W3C Cascad
Fontconfig will yield a matching system tha
Mozilla or other web browsers.**

(b) bold

*The original ideas for Fontconfig came about
Xft was originally designed to connect fonts r
Render Extension[render] to the X window sy
customization mechanisms, Xft included its o
creating yet another incompatible configurati
During a subsequent redesign of Xft, the conf
extracted and moved into a separate library
other applications. The development of Xft-b
Xft font selection. The need to embed the par
evident that a standard configuration file forr
XML[xml] would be a good fit for this task.
Development of Mozilla[mozilla] and Pango[p
about fonts during the selection process. Font
about every font in the system to aid in selec
inherited from Xft has proven effective and h
from the second version of the W3C Cascadin
Fontconfig will yield a matching system that
Mozilla or other web browsers.*

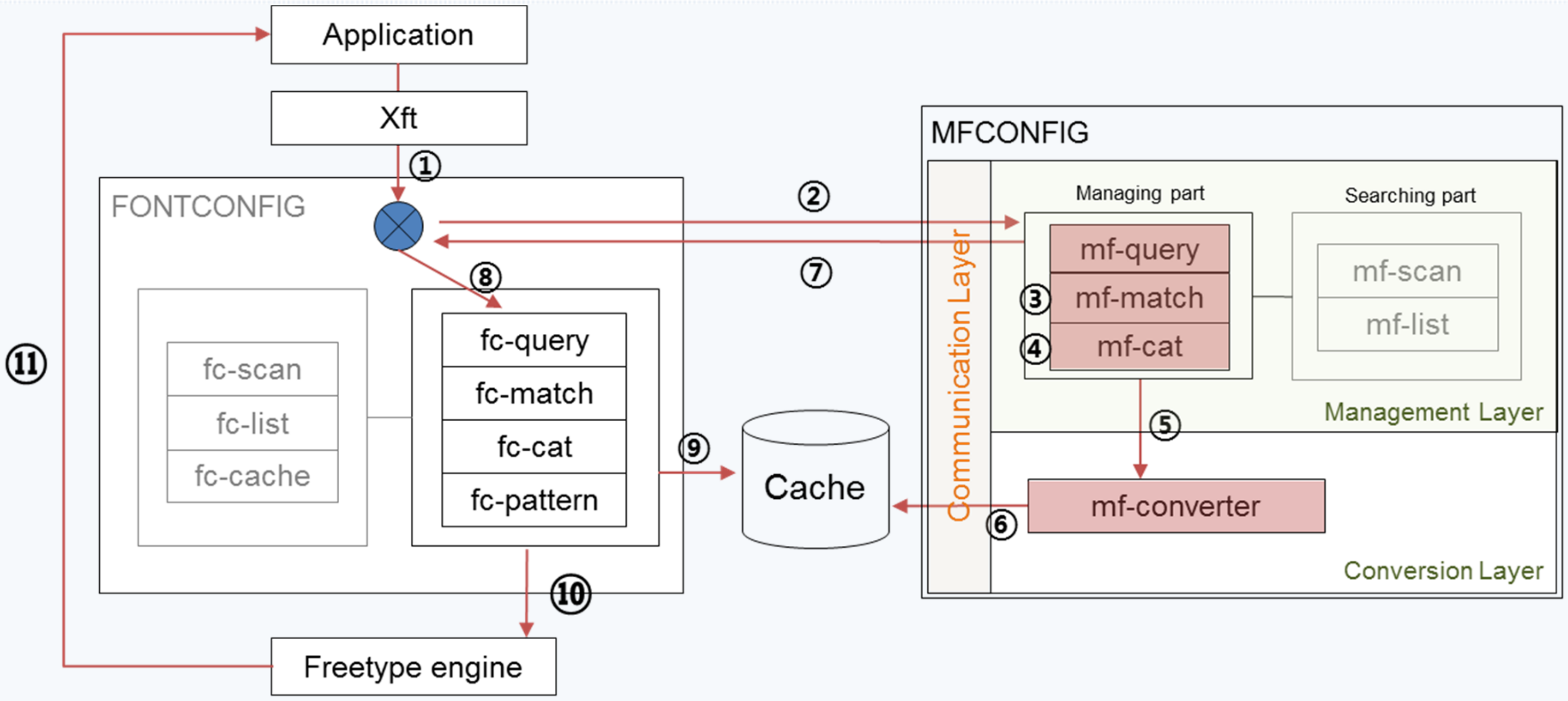
(c) italic

***The original ideas for Fontconfig came abou
Xft was originally designed to connect fonts
Render Extension[render] to the X window
customization mechanisms, Xft included its
creating yet another incompatible configura
During a subsequent redesign of Xft, the co
extracted and moved into a separate library
other applications. The development of Xft-
Xft font selection. The need to embed the p
evident that a standard configuration file fo
XML[xml] would be a good fit for this task.
Development of Mozilla[mozilla] and Pango[
about fonts during the selection process. Fo
about every font in the system to aid in sele
inherited from Xft has proven effective and
from the second version of the W3C Cascadi
Fontconfig will yield a matching system that
Mozilla or other web browsers.***

(d) bold-italic

04 EXPERIMENT *MFCONFIG* MODULE

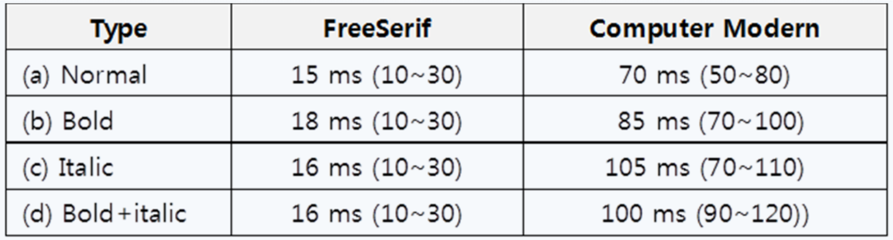
- **The performance analysis regarding the Freetype rasterizer:** **Processing time** **between** **requesting fonts with styles from application** **and** **successfully displaying fonts of on-screen**



04 EXPERIMENT *MFCONFIG* MODULE

Measurement time of test result

- Average time compare : 16 ms vs 90 ms



Type	FreeSerif	Computer Modern
(a) Normal	15 ms (10~30)	70 ms (50~80)
(b) Bold	18 ms (10~30)	85 ms (70~100)
(c) Italic	16 ms (10~30)	105 ms (70~110)
(d) Bold+italic	16 ms (10~30)	100 ms (90~120))

The Reason of time increment

- Need additional time for the conversion of the METAFONT font into corresponding font (outline font type)
- “mftrace” & “autotrace”

Thank you