Redesign of a METAPOST-based font generating system

Marek Ryćko, Bogusław Jackowski

Bachotek, 4 V 2019

A glance at history

A glance at history

- In 2001, the first release of the programmable METAPOST-based engine (METATYPE1), inspired by Janusz M. Nowacki, was released. Janusz was the first user and programmed several fonts using it: Iwona, Kurier, Cyklop, Antykwa Toruńska, and Antykwa Półtawskiego (the latter was later improved by GUST e-Foundry).

A glance at history

- In 2001, the first release of the programmable METAPOST-based engine (METATYPE1), inspired by Janusz M. Nowacki, was released. Janusz was the first user and programmed several fonts using it: Iwona, Kurier, Cyklop, Antykwa Toruńska, and Antykwa Półtawskiego (the latter was later improved by GUST e-Foundry).
- The popular font format of that time, PostScript Type 1, was usable in TEX (thanks to Tomas Rokicki's driver `dvips`), being a convenient replacement for the native TEX's PK bitmap format.

A glance at history

- In 2001, the first release of the programmable METAPOST-based engine (METATYPE1), inspired by Janusz M. Nowacki, was released. Janusz was the first user and programmed several fonts using it: Iwona, Kurier, Cyklop, Antykwa Toruńska, and Antykwa Półtawskiego (the latter was later improved by GUST e-Foundry).
- The popular font format of that time, PostScript Type 1, was usable in TEX (thanks to Tomas Rokicki's driver `dvips`), being a convenient replacement for the native TEX's PK bitmap format.
- METAPOST was not able to create PostScript Type 1 fonts, but it seemed to us that converting a set of EPS files generated by METAPOST to a PostScript Type 1 should be doable.

A glance at history

- In 2001, the first release of the programmable METAPOST-based engine (METATYPE1), inspired by Janusz M. Nowacki, was released. Janusz was the first user and programmed several fonts using it: Iwona, Kurier, Cyklop, Antykwa Toruńska, and Antykwa Półtawskiego (the latter was later improved by GUST e-Foundry).
- The popular font format of that time, PostScript Type 1, was usable in TEX (thanks to Tomas Rokicki's driver `dvips`), being a convenient replacement for the native TEX's PK bitmap format.
- METAPOST was not able to create PostScript Type 1 fonts, but it seemed to us that converting a set of EPS files generated by METAPOST to a PostScript Type 1 should be doable.
- Indeed, the using of fairly stable tools – METAPOST, `(G)AWK` and `T1Utils` package (by Lee Hetherington) – sufficed.

A glance at history

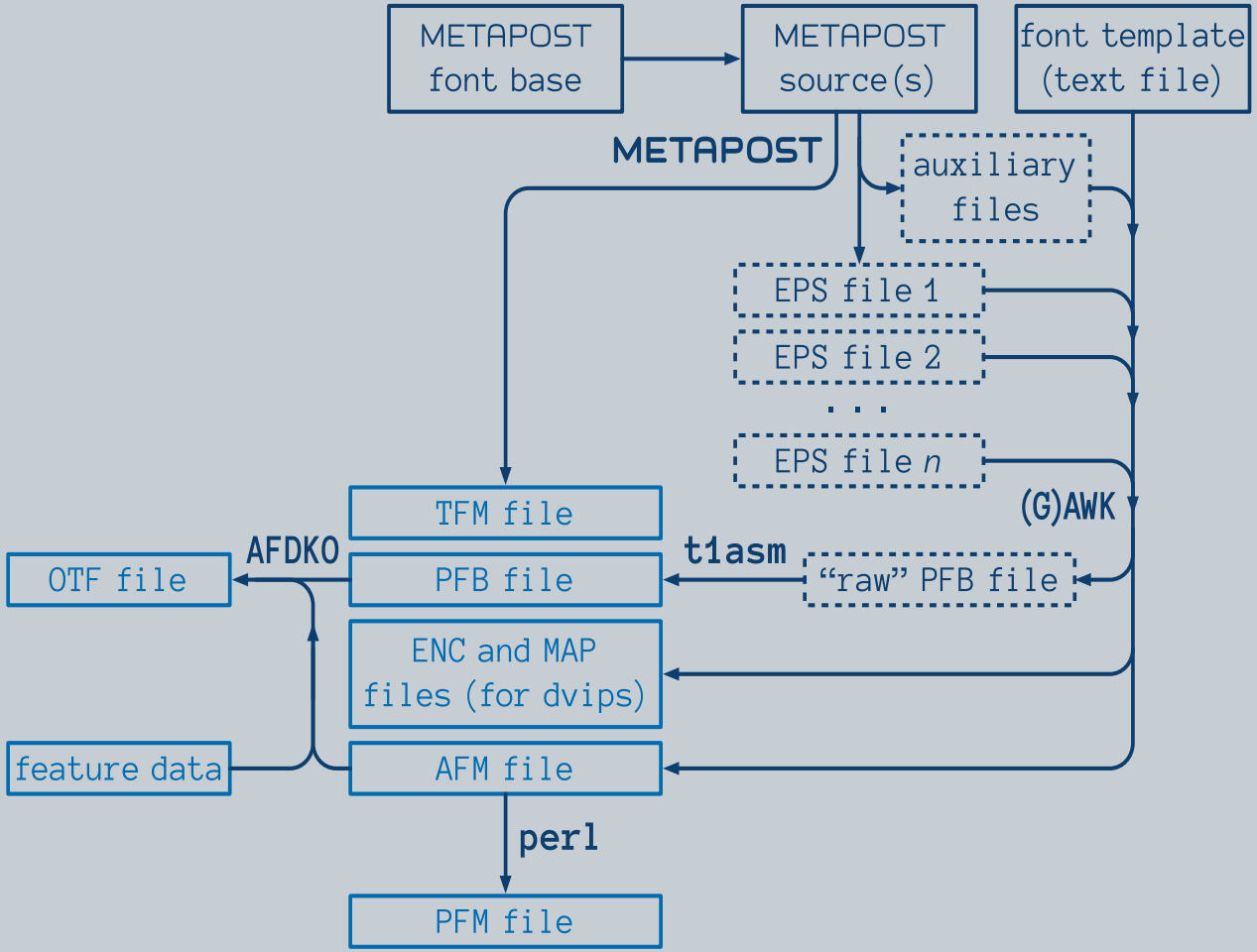
- In 2001, the first release of the programmable METAPOST-based engine (METATYPE1), inspired by Janusz M. Nowacki, was released. Janusz was the first user and programmed several fonts using it: Iwona, Kurier, Cyklop, Antykwa Toruńska, and Antykwa Półtawskiego (the latter was later improved by GUST e-Foundry).
- The popular font format of that time, PostScript Type 1, was usable in TEX (thanks to Tomas Rokicki's driver `dvips`), being a convenient replacement for the native TEX's PK bitmap format.
- METAPOST was not able to create PostScript Type 1 fonts, but it seemed to us that converting a set of EPS files generated by METAPOST to a PostScript Type 1 should be doable.
- Indeed, the using of fairly stable tools – METAPOST, `(G)AWK` and `T1Utils` package (by Lee Hetherington) – sufficed. For some period…

A glance at history

ENGINE 2001–2017 (B. Jackowski, J. M. Nowacki, P. Strzelczyk)

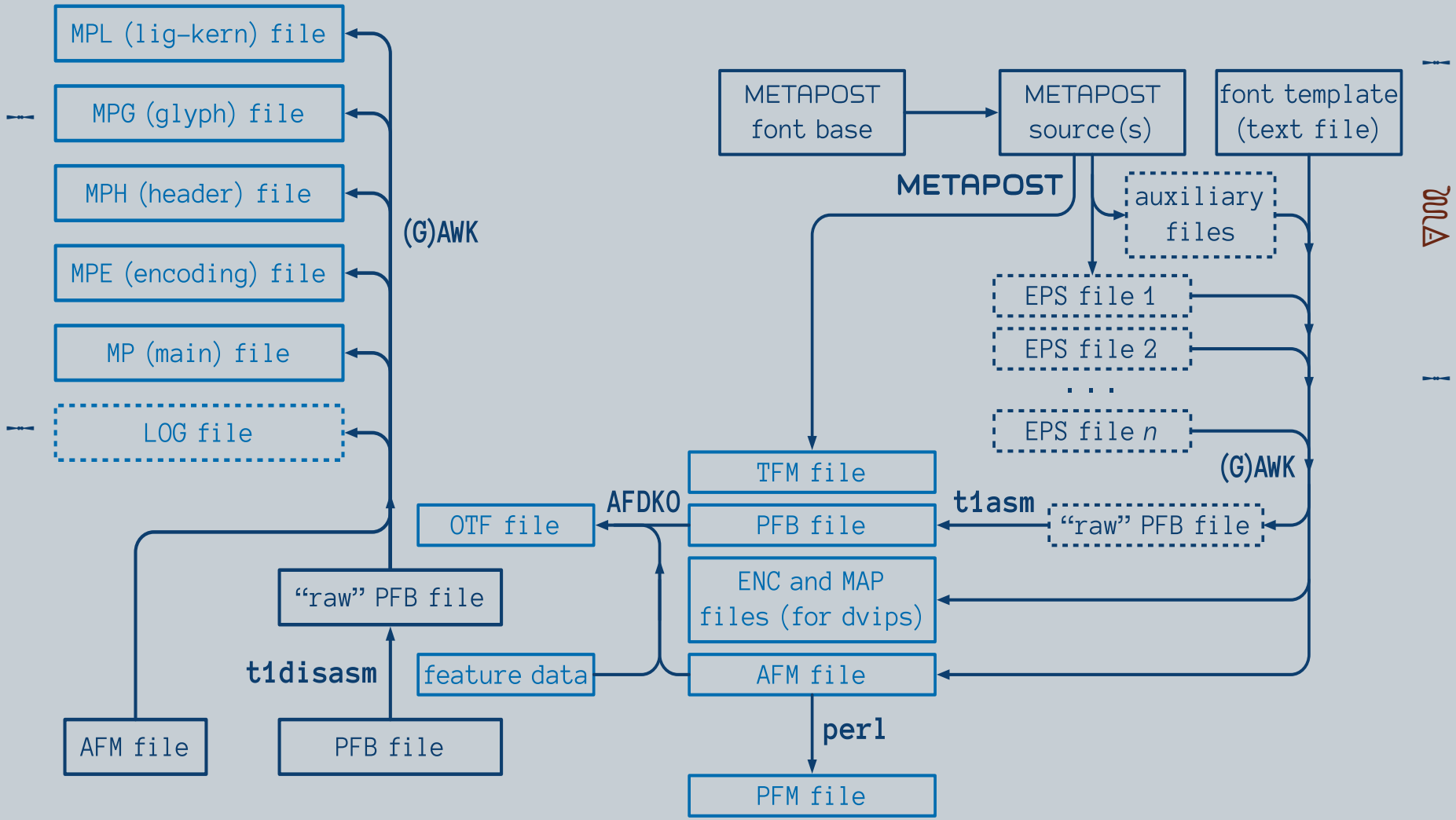
A glance at history

ENGINE 2001–2017 (B. Jackowski, J. M. Nowacki, P. Strzelczyk)



A glance at history

ENGINE 2001–2017 (B. Jackowski, J. M. Nowacki, P. Strzelczyk)



A glance at history

- As one could see, the engine from the very beginning was pretty complex, and it became even more complex due to the advent of OpenType font format (ISO Standard, 2007). As a result, Adobe font development kit for OpenType (AFDKO) was included and, moreover, `perl` was employed for generating binary printer metric files (PFM) for Windows.

A glance at history

- As one could see, the engine from the very beginning was pretty complex, and it became even more complex due to the advent of OpenType font format (ISO Standard, 2007). As a result, Adobe font development kit for OpenType (AFDKO) was included and, moreover, `perl` was employed for generating binary printer metric files (PFM) for Windows.
- We decided to extend the engine in order to make its output universal, not merely TEX-oriented.

A glance at history

- As one could see, the engine from the very beginning was pretty complex, and it became even more complex due to the advent of OpenType font format (ISO Standard, 2007). As a result, Adobe font development kit for OpenType (AFDKO) was included and, moreover, `perl` was employed for generating binary printer metric files (PFM) for Windows.
- We decided to extend the engine in order to make its output universal, not merely TEX-oriented.
- But the engine structure became non-uniform, scattered, and tiresome in installing and configurating. Hence a scarce number of users: Janusz M. Nowacki, Johannes Küster (splendid Minion Math), Jacek Kmiecik (Pyszczki/Faces), the GUST e-Foundry (TG Gyre, Latin Modern, and math companion to both, altogether 130 font files, more than 120,000 glyphs), and, perhaps, occasionally, somebody else.

A glance at history

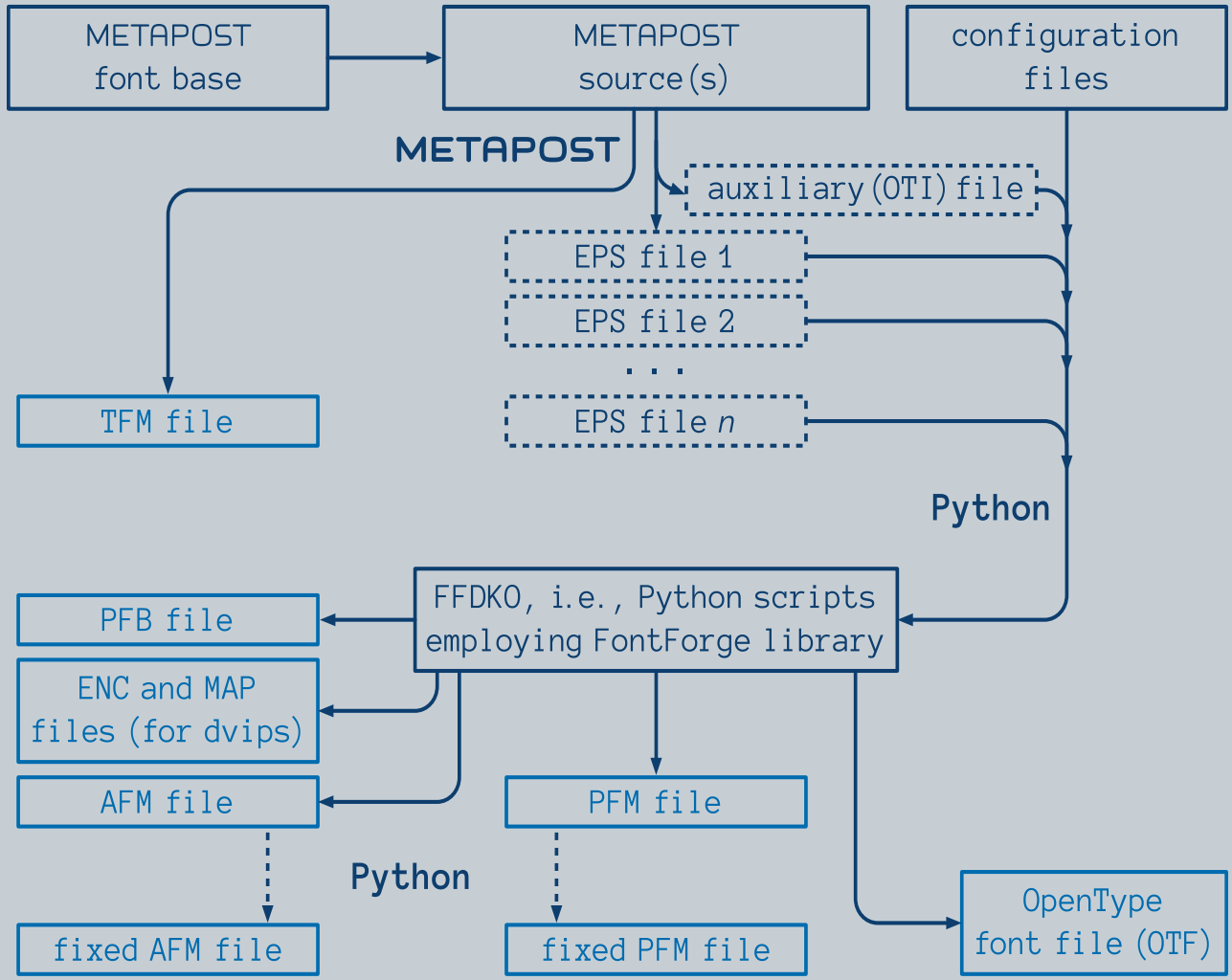
- As one could see, the engine from the very beginning was pretty complex, and it became even more complex due to the advent of OpenType font format (ISO Standard, 2007). As a result, Adobe font development kit for OpenType (AFDKO) was included and, moreover, `perl` was employed for generating binary printer metric files (PFM) for Windows.
- We decided to extend the engine in order to make its output universal, not merely TEX-oriented.
- But the engine structure became non-uniform, scattered, and tiresome in installing and configurating. Hence a scarce number of users: Janusz M. Nowacki, Johannes Küster (splendid Minion Math), Jacek Kmiecik (Pyszczki / Faces), the GUST e-Foundry (TG Gyre, Latin Modern, and math companion to both, altogether 130 font files, more than 120,000 glyphs), and, perhaps, occasionally, somebody else.
- Therefore, we finally decided to simplify the engine – we employed only METAPOST and Python (with FontForge library).

A glance at history

ENGINE 2018 (P. Strzelczyk, B. Jackowski)

A glance at history

ENGINE 2018 (P. Strzelczyk, B. Jackowski)



A glance at this moment

A glance at this moment

- Still, the system needed different shells for different operating systems.

A glance at this moment

- Still, the system needed different shells for different operating systems.
- Moreover, a user had to know much too much about the engine intestines in order to reconfigure or reinstall it.

A glance at this moment

- Still, the system needed different shells for different operating systems.
- Moreover, a user had to know much too much about the engine intestines in order to reconfigure or reinstall it.
- Therefore we attempted to simplify the engine further looking at it as a 4-structure construction: an input structure, an intermediate structure, an output structure, all governed by the configuration and installation structure.

A glance at this moment

- Still, the system needed different shells for different operating systems.
- Moreover, a user had to know much too much about the engine intestines in order to reconfigure or reinstall it.
- Therefore we attempted to simplify the engine further looking at it as a 4-structure construction: an input structure, an intermediate structure, an output structure, all governed by the configuration and installation structure.
- The main implementation idea is to use Python as a governing shell running METAPOST (which makes the engine independent of the operating system) and provide a simple yet universal installation (micro)language.

A glance at this moment

- Still, the system needed different shells for different operating systems.
- Moreover, a user had to know much too much about the engine intestines in order to reconfigure or reinstall it.
- Therefore we attempted to simplify the engine further looking at it as a 4-structure construction: an input structure, an intermediate structure, an output structure, all governed by the configuration and installation structure.
- The main implementation idea is to use Python as a governing shell running METAPOST (which makes the engine independent of the operating system) and provide a simple yet universal installation (micro)language.
- We hope that such approach will improve transferability of the engine – both to other users and systems.

A glance at this moment

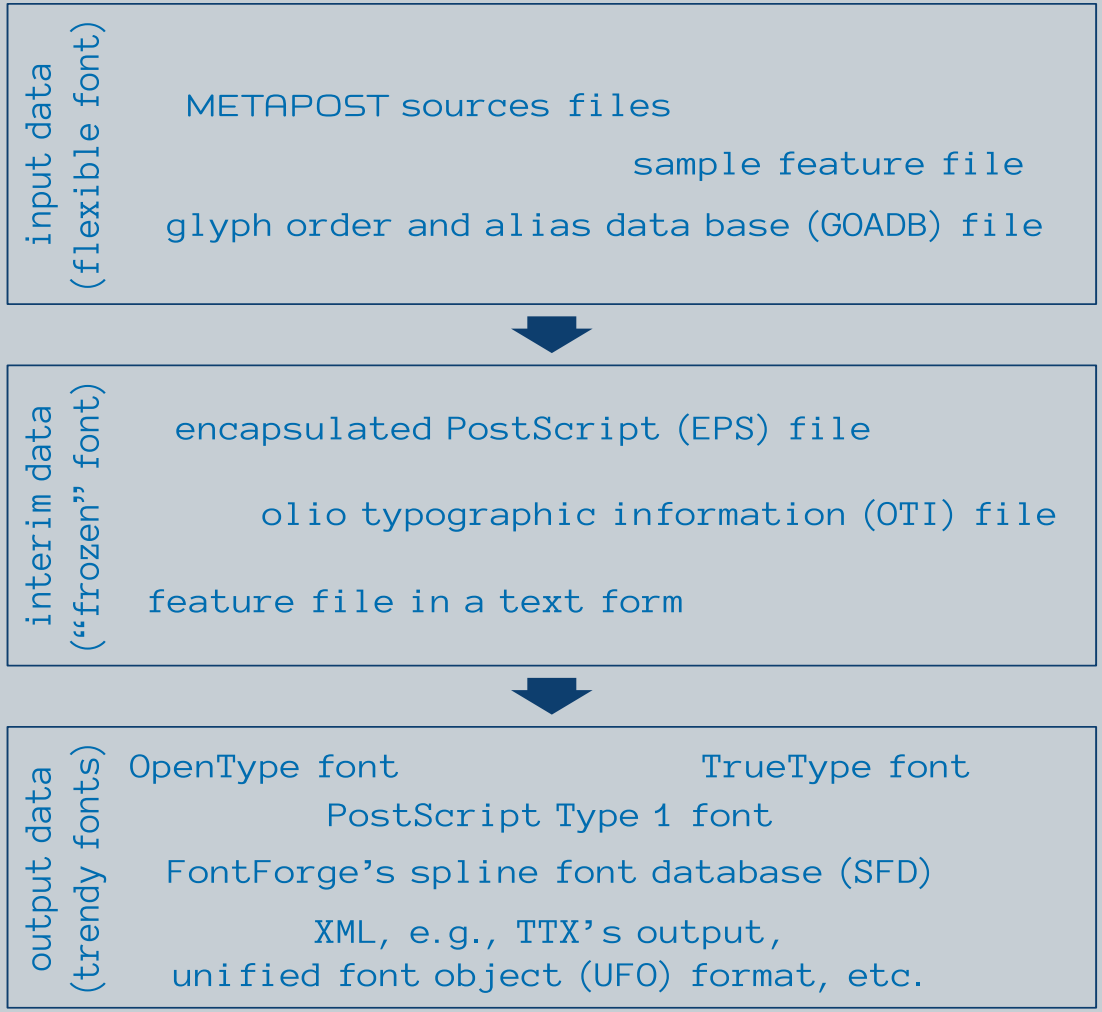
ENGINE 2019 (M. Ryćko, P. Strzelczyk, B. Jackowski)

A glance at this moment

ENGINE 2019 (M. Ryćko, P. Strzelczyk, B. Jackowski)

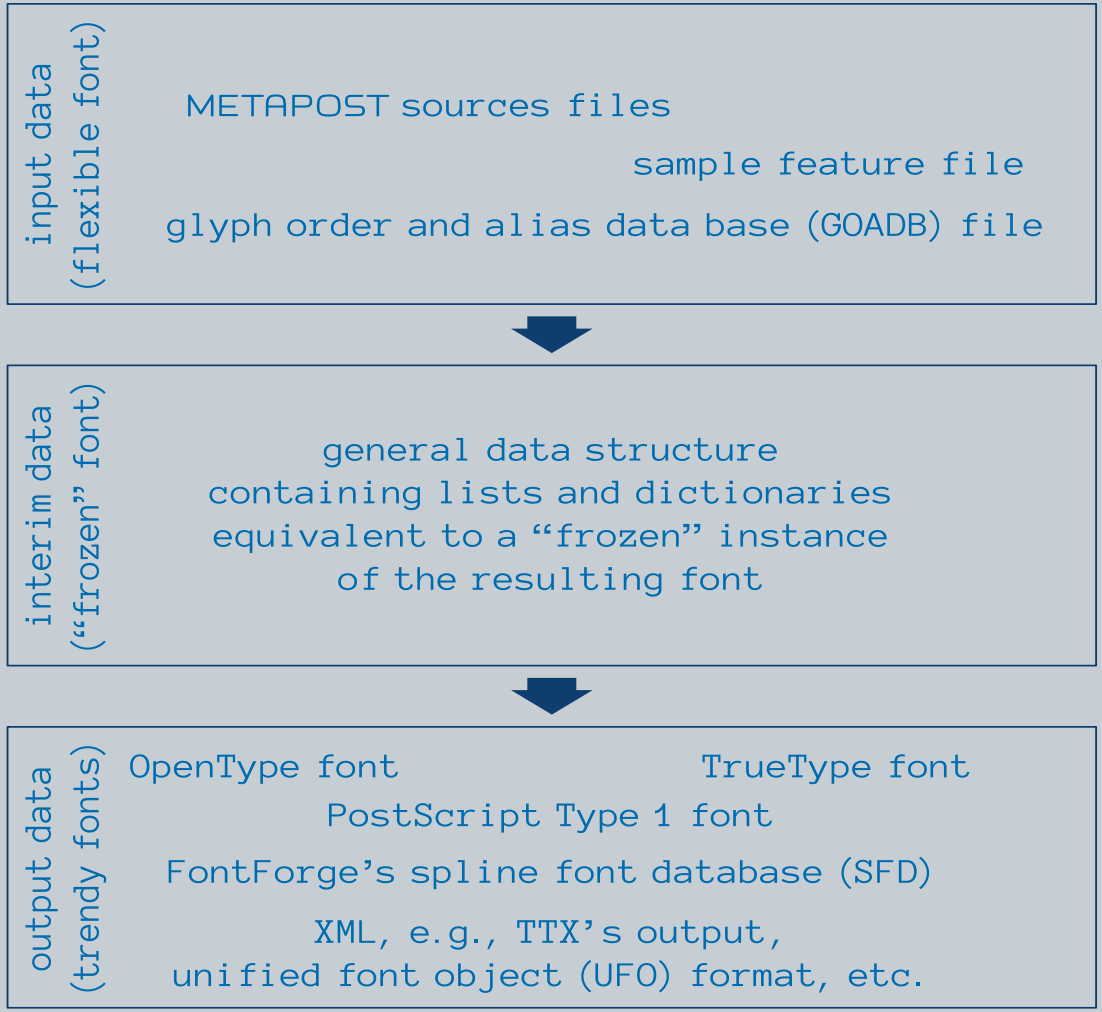
A glance at this moment

ENGINE 2019 (M. Ryćko, P. Strzelczyk, B. Jackowski)



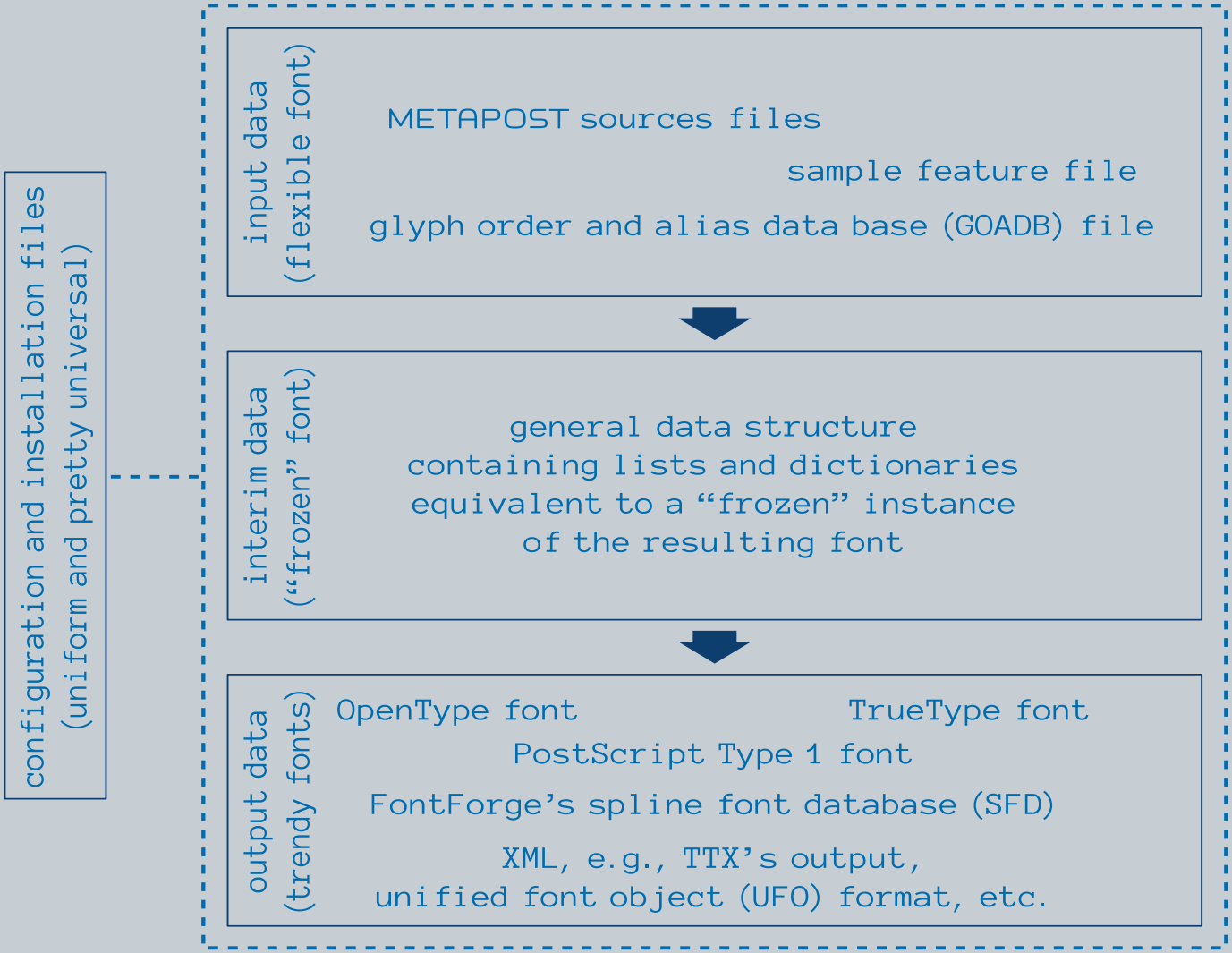
A glance at this moment

ENGINE 2019 (M. Ryćko, P. Strzelczyk, B. Jackowski)



A glance at this moment

ENGINE 2019 (M. Ryćko, P. Strzelczyk, B. Jackowski)



Future works

Future works

- In practice, the font generating engine cannot be as simple as depicted on the diagram which has been shown – two more components (or modules) are needed.

Future works

- In practice, the font generating engine cannot be as simple as depicted on the diagram which has been shown – two more components (or modules) are needed.
  - Important component is the converter which generates METAPOST sources out of the “trendy” fonts. The old engine was equipped by a converter from PostScript Type 1 fonts to METAPOST sources.

Future works

- In practice, the font generating engine cannot be as simple as depicted on the diagram which has been shown – two more components (or modules) are needed.
  - Important component is the converter which generates METAPOST sources out of the “trendy” fonts. The old engine was equipped by a converter from PostScript Type 1 fonts to METAPOST sources.
  - Another important component, not to say crucial, is the proofing module. The old engine employed for this purpose the `mft` utility from the standard collection of METAFONT-related programs provided by Donald E. Knuth.

Future works

- In practice, the font generating engine cannot be as simple as depicted on the diagram which has been shown – two more components (or modules) are needed.
  - Important component is the converter which generates METAPOST sources out of the “trendy” fonts. The old engine was equipped by a converter from PostScript Type 1 fonts to METAPOST sources.
  - Another important component, not to say crucial, is the proofing module. The old engine employed for this purpose the `mft` utility from the standard collection of METAFONT-related programs provided by Donald E. Knuth.

    (It is worth emphasizing that both converter and formater, although obsolescent, are still used by us).

Future works

- In practice, the font generating engine cannot be as simple as depicted on the diagram which has been shown – two more components (or modules) are needed.
  - Important component is the converter which generates METAPOST sources out of the “trendy” fonts. The old engine was equipped by a converter from PostScript Type 1 fonts to METAPOST sources.
  - Another important component, not to say crucial, is the proofing module. The old engine employed for this purpose the `mft` utility from the standard collection of METAFONT-related programs provided by Donald E. Knuth.

    (It is worth emphasizing that both converter and formater, although obsolescent, are still used by us).
- We have the plan to finish the project of the TEX Gyre font enhancement as soon as possible (we recall that so far two fonts were enhanced: TG Adventor and TG Pagella), including TEX Gyre math fonts.

Dreams

Dreams

- We do not give up our interest in fonts. But some day we will do… Our basic dream is to pass the font generating system on to people who like programming and font creation. Currently, as the experience shows, it is hardly transferable. We guess that the reason is its intricate structure and therefore the system is perplexing in use.

Dreams

- We do not give up our interest in fonts. But some day we will do… Our basic dream is to pass the font generating system on to people who like programming and font creation. Currently, as the experience shows, it is hardly transferable. We guess that the reason is its intricate structure and therefore the system is perplexing in use.
- We hope that the current redesign will definitely help to cure the mentioned drawbacks, in particular, the system will substantially become more efficient. And there are a lot of splendid fonts that could be processed, e.g., equipped in math…

Dreams

- We do not give up our interest in fonts. But some day we will do… Our basic dream is to pass the font generating system on to people who like programming and font creation. Currently, as the experience shows, it is hardly transferable. We guess that the reason is its intricate structure and therefore the system is perplexing in use.
- We hope that the current redesign will definitely help to cure the mentioned drawbacks, in particular, the system will substantially become more efficient. And there are a lot of splendid fonts that could be processed, e.g., equipped in math…
- Our far-reaching dream is to work out a “typeless” system for setting documents. We’d like to replace the notion of frozen font (so called “variability” does not help too much) with a hierarchical collection of programmable objects, mutually independent but able to comunicate with each other.

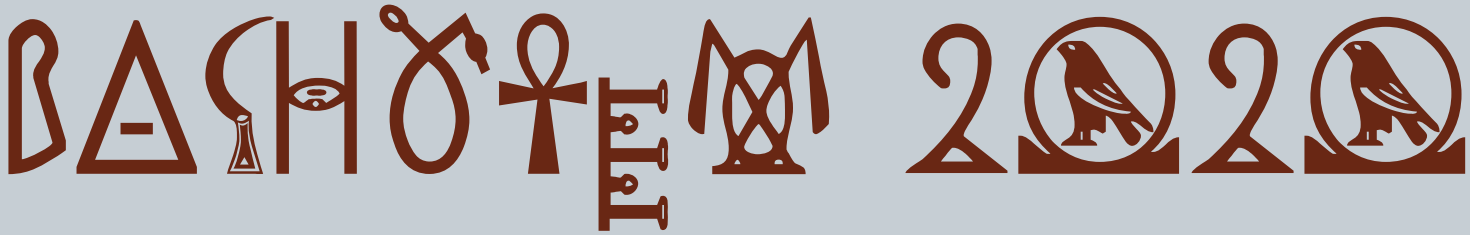
Dreams

- We do not give up our interest in fonts. But some day we will do… Our basic dream is to pass the font generating system on to people who like programming and font creation. Currently, as the experience shows, it is hardly transferable. We guess that the reason is its intricate structure and therefore the system is perplexing in use.
- We hope that the current redesign will definitely help to cure the mentioned drawbacks, in particular, the system will substantially become more efficient. And there are a lot of splendid fonts that could be processed, e.g., equipped in math…
- Our far-reaching dream is to work out a “typeless” system for setting documents. We’d like to replace the notion of frozen font (so called “variability” does not help too much) with a hierarchical collection of programmable objects, mutually independent but able to comunicate with each other.
- In our final words, we’d like to wish ourselves good dreams!

That’s all, so far… – thank you for your attention!

That's all, so far... – thank you for your attention!

And see you next year!



CREDITS

The funny "Egyptian" symbols that we used as a fancy font for BachoTEX's printed gadgets were designed by Petra Heidorn (pen name CybaPee Creations). The collection of her free fonts can be downloaded from Luc P. Devroye's web site http://luc.devroye.org/fonts-30503.html