

Table of Contents

1	Galápagos Design Group	2
1.1	Mission statement	2
1.2	History	2
1.3	Staff, employees	3
1.4	Working fields.....	4
2	Type.....	6
2.1	Terms	6
2.2	The design process	7
2.3	Kerning.....	10
2.4	The designer's tools.....	12
2.5	Tests	13
2.6	Type nostalgia, set in my own type	14
3	Hinting.....	16
3.1	Font Scaling	16
3.2	Coordinate Rounding, gridfitting	16
3.3	TrueType and Type 1 hints	17
4	OpenType	19
4.1	OpenType Layout.....	19
5	Content Management	21
5.1	Some rules to manage content	21
5.2	HTML versus XML	21
5.3	Perl scripts by myself	22
6	Result	26
7	Appendix.....	27
8	Bibliography	28

1 Galápagos Design Group

1.1 Mission statement

“The mission of Galápagos Design Group is to produce high-quality, practical type products to deliver creative and innovative design services and to contribute to the worldwide aesthetic and technical advancement of type.”¹

1.2 History

Galápagos Design Group was incorporated in January of 1994. Conceived by a core group of experienced and talented type designers and type technologists, the company now consists of a formidable ensemble of design and type professionals, with offices in Littleton, Massachusetts. With an average of 20 years of type design experience, the senior staff has been providing the highest quality type products and services to the graphic arts, printing, and electronic publishing industries since the inception of the company.

Most designers have worked at Bitstream, a type company which was founded in 1981 as “the first digital font foundry”. Bitstream has provided quality products and innovative tools for the graphics communication industry. As a leading developer of font technology and digital typefaces, Bitstream has set a standard for quality, which is reflected in a extensive library of Latin and international fonts. Their commitment to value is shown in a wide range of products for font management, font rendering, and typography for the Internet.

Out of this thriving company, Larry Oppenberg, Dennis Pasternak, George Ryan, Stevenson Zafarana, Pamela Grant and Geoffrey Greve started their own business which they named Galápagos Design Group.

Just as the islands of their namesake represent a new view of an old world, so too does Galápagos, the company, represent a new approach to the traditional world of typography. Combining the traditional with the original, aesthetics with technology, form with content, Galápagos Design offers a wide range of products and services, including both popular and custom solutions.

¹ www.galapagosdesign.com

A broad spectrum of state-of-the-art design tools, including sophisticated off-the-shelf applications, as well as powerful, integrated proprietary design systems, allow Galápagos designers to perform rapid prototyping and feasibility analyses, and to deliver in a timely and economical manner.

1.3 Staff, employees

The President and senior staff of Galápagos Design Group bring talent, experience and commitment to bear on the challenges of a rapidly evolving marketplace. Together, they have made significant contributions to the type libraries and products of a number of the most respected companies in the business.

Being my tutor and supervisor, some lines follow about Dennis Pasternak, who together with Larry Oppenberg made this internship possible for me.

With nearly two decades of experience in the study and practice of type design, Dennis is an accomplished designer of



original typefaces. Unlike many who work in the digital medium, Dennis considers himself a craftsman, in the tradition of the masters who worked prior to the digital era. Although it is mathematical figures and patterns that populate his computer monitors, Dennis often senses the wood, metal and brushes employed by his professional antecedents.

Before becoming one of the founders of Galápagos Design Group in 1994, Dennis worked for several of the major players in the type arena, including Compugraphic Corporation, Autologic and Bitstream

Dennis' growing body of original designs include the typefaces developed during his tenure at Galápagos Design Group – the Maiandra family (part of is now shipped with Windows 98; also I have seen the typeface on the soccer playfield), Baltra and Stylus. A noteworthy earlier design is the text/display family, Bitstream Chianti. All of Dennis' designs have traditional roots, and are highly readable; they focus on the reader, rather than on their potential as graphic elements. When I came to Galápagos he just finished a font for the Dunkin' Donuts chain, which they now use in their ads all over the states. At this job two

different fonts had been “interpolated” into the new one. Dennis holds a Bachelor of Fine Arts degree in design from the Massachusetts College of Art. He is also a member of the Association Typographique Internationale (ATypI).

1.4 Working fields

1.4.1 Type

There is a variety of different tasks the group fulfills working with type. They develop new typefaces from scratch, do conversions from one font-technology to the other, or just improve existing fonts. To see the complexity of the work, I present the thorough evaluation of fonts which is necessary before starting to work on existing typefaces: You have to check whether there are some glyphs² missing or if they are in an incorrect position. Possibly many glyphs themselves need work, that is, the outline must be altered, or you just have to shift the glyph left or right. Accents bring a lot of problems: they could be de-positioned or even have discrepancies in design, that is the design doesn't match the prime glyph/accent-design. Also you might have to change the characters' unit values.



1.4.2 Web

It is Steve Zafarana who maintains Galápagos' Website and offers web design for exterior companies.

Steve has over 20 years experience as a type designer and artist. Steve holds a certificate in Fine Art, as well as one in Commercial Art, both from the Monserratt School of Art. He also studied film animation at the Carpenter Center.

² An image for a character in a particular font and style.

Steve's interest in illustrations and cartoons is reflected in some of his original type designs, which include the cartoon lettering font ITC Fontoon, the whimsical Fontoonies, Gargoonies, and Backyard Beatsties. Steve is also the designer of the controversial Safefont, as well as the more traditional typefaces Tangient and Wakefield.

1.4.3 Technology

All this work is done with the following technical equipment. Galápagos Design Group runs a NT Server, a SUN Enterprise and mostly Macintosh machines. From my point of view it does not make much difference which platform is used since most of the font tools are available on both the Mac and PC side. There is X-Terminal software to use the proprietary tools running on SUN-system via the workstations. For some unusual jobs, there are even some old alpha machines available.



George Ryan (type designer and system administrator) and Dennis Pasternak (type designer). Dennis working with his favorite tool: Fontographer

2 Type

2.1 Terms

2.1.1 Typefaces and fonts

In modern usage the terms font, typeface and family are often confused.

Font

Traditionally, the term "font" represents a complete set of characters or symbols, which share the same size and style. For example, 12 point Goudy Oldstyle Bold is a font. Fonts can be as small as the basic alphabet or up to hundreds of characters. Some languages, like Japanese, can exceed these numbers, which make them more difficult to access from the standard keyboard. Originally derived from the word "found" as in typefoundry.

Typeface

Often named after a designer, a typeface or "face" (e.g. Goudy Oldstyle) is an interpretation of a character set that shares a similar appearance and design. The character set includes letters, numbers, punctuation and symbols. On computers, "typeface" is used interchangeably with the term "font".

Family

All the type sizes and styles of one typeface. A complete character set of a font. The group shares a common design but can differ in attributes such as character width, weight, and posture (i.e., Roman vs. Italic).

Style

A visual variation of a basic typeface used to create emphasis. Type style is important since it can attract (or repel) the reader's eye. The four basic computer styles are Plain, Bold, Italic, and BoldItalic.

2.1.2 Legibility and readability

As stated above Dennis creates highly readable typefaces – but what defines readable?

Legibility and readability are separate, though connected, aspects of type. Properly understood, and used in the meanings appropriate to the subject, the two terms can help to describe the character and function of type more precisely than legibility alone.

Legibility means the quality of being easy to read. In typography it means the

quality of being decipherable and recognizable. It's the term to use when discussing the clarity of single characters. It is a matter for concern in text sizes. In display sizes legibility ceases to be a serious matter; a character which causes uncertainty at 8 point will be plain enough at 24 point.

Readability in typography describes the quality of visual comfort: if the columns of a newspaper or magazine or the pages of a book can be read for many minutes at a time without strain or difficulty.

2.2 The design process

Let's start with an historic example by Harry Carter³:

The success or failure of a type is very much a question of getting a good balance of white inside and outside the letters. The interior areas of letters are fixed by the shape of the letters, but the spaces at each side of them are at will: at the will of the type designer. The 'fitting' of letters – the allocating of the correct amount of space to each side of them, so that when they are associated into words they have a balanced relationship, without unsightly gaps or congestion – is a process fundamental to the success of a type design.

In his "WAD to RR" (see Appendix D), written in response to Rudolph Ruzicka's⁴ request for advice on type designing, W. A. Dwiggins⁵ said, 'Each type letter, wherever it goes, carries along with it two /fixed/ blank spaces, one on each side. And of course, each one of the 26 is likely to be placed alongside any one of the other 25 with /their/ fixed blank spaces ... the letter shapes occur in groups of similars: when you have solved for n alongside of n you are close to a workout for h l j l m and for the stem sides of b d k p q – a proper fitting for o gives you a line on the round shapes ... a, c, e, on their open sides, and f g r t are hard to fit ...' 'There isn't any fitting formula worked out yet. Griffith⁶ says there can't be any: that it is a job for the eye alone.'

³ Harry Carter, Britain 1901 – 1982, lawyer, linguist, book and type designer and peerless typographic historian; father of Matthew Carter.

⁴ Rudolph Ruzicka, Czechoslovakia 1883 – 1978, wood-engraver and book designer active over the first three-quarters of the 20th century, associated with C.H. Griffith and Jackson Burke at Mergenthaler Linotype designing type after the Second World War.

⁵ William Addison Dwiggins, USA 1880 – 1956, Leading American book designer of the first half of the 20th century, closely associated with C.H. Griffith, and leading designer for the Mergenthaler Linotype Company from 1929 until after the war.

⁶ Chauncey H. Griffith, USA 1879 – 1956, Kentucky printer and Linotype salesman who directed the growth of the Linotype library from 1915 to 1948, improved with his fonts the look of the world's newspapers.

2.2.1 Groups of letters

Most of the letters are formed of straight strokes or round strokes, or a combination of them; and the direction of emphasis is vertical. The letters can be grouped like this:

Letters with straight upright stroke:

B D E F H I J K L M N P R U b d h i j k l m n p q r u

Letters with round stroke:

C D G O P Q b c d e o p q

Triangular letters:

A V W X Y v w x y

The odd ones:

S T Z a f g s t z

Out of this grouping there are a few letters which are called control characters: The capital H, D and O, the lowercase , o and p and the figure 0. These key characters represent the round and square bodies and strokes and a combination of the two. They also show the proper round and square height relationships above and below the baseline.

2.2.2 Character spacing

This selection of letters is used for character spacing. For the spacing of the capitals the basis is the H. The designer will have tested it with a few other letters and satisfied himself that the weight of its strokes, the width of its interior space, and the shape and length of its serifs are all in accord with his purpose.

When setting together four Hs in a row, their eight upright strokes are equally spaced. They might look good for a bold face, but for a normal weight and fairly broad interior whites the crossbar of H acts as a ligament drawing the upright together, so the side spaces must be reduced until they look equal to the interior spaces. Creating this illusion is a task for the eye and careful judgment. Time is needed to get the balance right. It is time well spent, because much depends on it. The serifs at the four corners of H make a linking effect with adjacent letters. Because they are absent in a sans-serif type the side-spaces of a n sans-serif H have to be narrower than those in a serified face.

When the Hs look harmonious, the spaces between them not too open, not too cramped, the distance between them is measured. A half of the amount is now

the appropriate allocation for each side of H, and for all other capitals having a straight vertical stroke.

The next letter to deal with is O, which is positioned between two pairs of correctly-spaced Hs, and the spaces on each side of it are reduced or increased until all five letters seem to be in balance. Then, two Os using that side space are placed side by side between the two pairs of Hs, thus, HHOHH. This test may show the need to revise the fitting of O – and even the H, because it is a new test of that character too. When that has been done the process has reached the stage when the ideal spacing of the two most useful capitals, the “standards”, has been achieved.

2.2.3 The capitals

With the spacing of H and O now known precisely, it is possible to finish the drawing of the rest of the alphabet, by taking note of the groups of letters listed earlier and spacing them as indicated here:

d A d	a B c	e C c	a D e	a E c	a F c
e G b	a I a	d J a	a K d	a L d	b M a
b N b	a P e	e Q e	a R d	d T d	a U b
d V d	d W d	d X d	d Y d	c Z e	

S must be spaced visually, between standards

where ...

- a Same as H
- b Slightly less than a
- c About half of a
- d Minimum space
- e Same as O

2.2.4 The lowercase

The same method is used for the fitting of the lowercase, the standards being the n and o. The width between the uprights of n is measured, and a half of that amount is given to the left side of the letter slightly less on the others side, because the arched corner seems to add to the space. Four ns are set in a row so that the eight uprights look equally spaced. Then the lowercase o can be dealt with, and checked in combinations with n, like this:

nnonn nnonon nnoonn

With the n and o established as standards, the rest of the alphabet is organized

by this scheme:

a b e	e c f	e d a	e e f	c h b	c i a
a j a	c k d	c l a	a m b	c p e	e q a
a r d	b u b	d v d	d w d	d y d	

a f g s t z must be spaces visually, between standards

where ...

- a same as left side of n
- b same as right side of n
- c slightly more than left side of n
- d minimum space
- e same as o
- f slightly less than o

2.3 Kerning

In traditional metal typography, a kern is the part of a letter that extends beyond the left or right edge of the rectangular type body. Some automatic typesetting machines could handle kerned type, but others forced all the character to be within the rectangular body. The Linotype was one of the latter, resulting in many inelegant italic letters – it's the cause of Sabon's many admirable workarounds. Being fragile, kerns could break off if the type was mis-handled.

In digital typography, kerning has a different meaning. The old worry about fragility has disappeared; so the italic *f* can keep its grace. Digital kerning (or, more precisely, "pair kerning") allows the spacing between any pair of characters to be specified, allowing, for example, an *r* following a *T* to nestle underneath the right-hand bar slightly, or an *LY* pair to nudge closer together. It is normally the task of the type designer to devise all the kerning values. In the days of metal, these adjustments were only possible with extreme labor, and were almost never seen.

In TrueType, pair kerning has always been possible in the kern table, where character combinations are stored with the amount (in font design units) to shift the second character when it comes after the first. Not all applications bother to use kerning information, so the default rectangular body should always be very carefully controlled. Triple-kerning (such as for the occasionally troublesome combinations *fff*, *ffl*, *ffi*, ...) is supported in Open Type.

Dennis Pasternak tries to avoid kerning as far as possible: “Less is more: A good font doesn’t need much kerning, as long as the widths are set carefully!”

2.3.1 Common kern pairs

A table of the most common 295 kern pairs:

AC	AG	AO	AQ	AT	AU	AV
AW	AY	AÓ	AÕ	Ap	Au	Av
Aw	Ay	BA	BU	B,	B.	CA
C,	C.	DA	DV	DW	DY	D,
D.	FA	Fa	F,	Fe	Fi	Fo
F.	Fr	G,	G.	JA	Ja	J,
Je	Jo	J.	Ju	KO	Ke	Ko
Ku	Ky	LT	LV	LW	LY	LÓ
LÕ	Ly	NA	N,	N.	OA	OT
OV	OW	OX	OY	O,	O.	PA
Pa	P,	Pe	Po	P.	QU	Q,
Q.	RO	RT	RU	RV	RW	RY
S,	S.	TA	TO	Ta	T:	T,
Te	Th	T-	Ti	To	T.	Tr
T;	Tu	Tw	Ty	UA	U,	U.
VA	VG	VO	Va	V:	V,	Ve
V-	Vi	Vo	V.	V;	Vu	WA
WO	Wa	W:	W,	We	Wh	W-
Wi	Wo	W.	W;	Wu	Wy	YA
YO	YS	Ya	Y:	Y,	Ye	Y-
Yi	Yo	Y.	Y;	Yu	ab	ag
ap	at	av	aw	ay	bb	b,
bl	b.	bu	bv	by	c,	ch
ck	cl	c.	cy	c space	,n	,Ó
.Õ	dd	dv	dw	dy	eb	e,
eg	ep	e.	ev	ew	ex	ey
fa	f,	fõ	fe	ff	fi	fl
fo	f.	fÓ	fÕ	ga	g,	ge
gg	gi	go	g.	gr	gy	hy
iv	ke	ko	ky	lw	ly	mu
my	nv	nu	ny	o,	og	o.
ov	ow	ox	oy	p,	p.	py
.Ó	.Õ	. space	nÒ	ÒÔ	ÔÔ	nÔ
Ó space	ÕÓ	ÕÕ	Õd	Õl	Õr	Õs
Õ space	Õt	Õv	ra	rc	r:	r,
rd	re	rg	r-	ri	rk	rl

r m	r n	r o	r p	r .	r q	r r
r s	r ;	r t	r u	r v	r y	s ,
s .	s w	; space	space A	space Ò	space Ô	space T
space V	space W	space Y	v a	v ,	v e	v o
v .	w a	w ,	w e	w h	w o	w .
x e	y a	y ,	y e	y o	y .	z e
z o						

2.4 The designer's tools

The problem with most of the programs listed below is that they don't deal natively with TrueType. Instead, as they load the font, they convert the outlines into PostScript-style cubic Bézier curves, and discard all the hints. For high quality fonts at low resolution, this is a tragic loss. TrueType hinting takes the form of little programs attached to each glyph, and it is admittedly hard, in fact virtually impossible, to work out automatically which program instructions can remain, and which must change when a glyph is modified. However, it's a shame that you can't leave alone any glyphs you don't modify: these programs affect every glyph.

2.4.1 Fontographer

Fontographer originated on the Macintosh in 1986. It will let you draw with a mouse or a digitizing tablet and create PostScript, TrueType, or EPS files. The program was developed by Altsys and later acquired by Macromedia. Both of these companies have updated Fontographer regularly and it has been adopted by many famous type designers as their primary design tool. The designers at Galápagos use it as main tool for designing the outlines.

One key advantage is its ability to import Freehand and Illustrator files, so designers can create a design in one of those programs if they wish and then complete the work and generate the actual typefaces in Fontographer. The program will also trace scanned images automatically to create usable characters; morph two designs together; create Multiple Master typefaces; add automatic spacing, kerning, and hinting to any typeface produced. These auto-hinted fonts can be greatly improved with a dedicated hinting system, though.

2.4.2 RoboFog

This "typehacktool" released by Peter van Blokland is a fully scriptable version of Fontographer 3.5 (Mac), which allows you now to write programs in the Python

language to control all aspects of font design -- scripts for many tasks are available, for example to automate the process of positioning the accents upon the base character.

2.4.3 FontLab

FontLab originated in Russia and was brought to the United States in 1994, It's fully professional and is the first typeface design program that was developed exclusively for Windows. Sold outside Russia by Pyrus North America, it has some enviable features. You can create a library of parts of characters for reuse, use multiple layers in creating a character, and move any point on a curve without using a node or control point. Like Fontographer, it supports scanned images, morphing, automatic kerning and hinting, and output in PostScript, TrueType, and EPS formats. It has an macro language that lets you develop your own special-effect routines for changing the weight of the outline for example.

2.4.4 Kernus

Kernus is a kerning utility that is available only for the Macintosh. With Kernus, you can set how tight you want the kerning to be, and how many kerning pairs you want. If you select a number like 500 or less, Kernus does the kerning from special character-combination lists) that include only the most important kerning pairs. The lists can be created by yourself as well.

2.5 Tests

Once you've designed the basic characters, you can run many tests by simply printing out different test-pages using the new font.

The individual glyphs have to be printed out in different sizes, some art-words like "hamburgevons" are used to have a look at the color of the font. Color, which is the overall gray-level of one typeface, might be spoiled occasionally by bad design, one letter looking too dark or too light compared to the others. It is more often spoiled by bad spacing, letters being crammed together in one place and set loosely in another. Either of these conditions create blotches of a different intensity in a word or page, and these blotches distract legibility.

A table showing all characters using a standard-font like Helvetica next to the letter using the new font is printed out to see if every glyph is in the correct position.

2.6 Type nostalgia, set in my own type

The days are over where the type-designer has drawn his letters on huge sheets of paper, which then were photographically reduced to make masters for photocomposition machines. The drawings were made to scale about 100 times actual



size. Using a pantographic routing machine, where the drawing of a character was placed at one end of the machine's table, at the other end a wax-coated metal plate was locked in position. The operator carefully tracked the follower point along the lines

of the drawing, and the cutter at the end of pantograph arm imitated the movement, in reduction, on the waxed plate.

After that, the plate was electrolytically coated with copper. The copper shell was backed with lead, and the result was an accurate representation of the character, about one quarter of the size of the drawing, and in firm relief; it was



called a pattern. These patterns were used to manufacture punches, using a punch-cutting-machine also with a kind of pantographic principle. The punch had to be

hardened to be used in the matrix stamping press finally. So for the first time the type design now existed on paper in the form of accurate drawings, mirrored-backwards though.

Many of these drawings can be seen in the Museum of Printing⁷ in North Andover, Massachusetts. The complete Linotype and Mergenthaler collection resides under its roof. Included are drawings by Larry Oppenberg, my boss at Galápagos Design Group, and George Ryan, now spending much time on system-administration at Galápagos apart from type-design. This collection being the biggest in the world of its kind was un-understandingly rejected by famous art-museums. So Larry, Chris and myself spent many Saturdays recompiling the collection in the Museum of Printing in Andover. We had to sort the drawings by typeface family and -size, which overall was a quite interesting task, being able to look at these wonderful drawings.

This chapter is set with my own typeface. Of course one cannot say that it is a nice and good-looking, maybe not even working font concerning readability. But looking at the fact that I just had multiple weeks I am satisfied with the outcome. Let's keep in mind that the designers at Galápagos Design Group all come with more than 25 years of experience in type-design.

I didn't start to design this font right away. To get a feeling for the tools I digitized drawings by Steve Zafarana for his new picture font "Fontoonies". It's hard to say whether it is more time consuming to redraw the



outline with the mouse or to correct the result of auto-tracing scanned bitmaps. My own font I started – as described above – with the capitals H and O, the lowercase n and o. The other characters are designed using more or less parts of the control characters. My progress is documented in the Appendix B pages.

⁷ www.museumofprinting.org

3 Hinting

Hinting is a very technical stage of the modern font design process. It is the stage when art finally meets technology. A lot of technical information and much experience is needed to make well-hinted fonts.

3.1 Font Scaling

One of most important features of outline fonts is that they can be used on many different output devices – from displays to imagesetters. Because character outline shapes are defined as sequences of lines and curves it is easy to scale outlines to any size and resolution. However, almost all output devices have discrete elements arranged in a regular rectangular raster and the images that these devices produce are constructed using discrete cells. Each cell in an output image has integer coordinates and is called a “pixel” (picture cell).

To measure scaled outlines in a resolution-independent way, it is convenient to define a quantity called Pixels Per eM (PPM). This is the number of pixels that can be fitted in the font’s height. Font height is a basic font unit equal to the Units Per Em (UPM). In TrueType fonts UPM is usually equal to 2048 and in Type 1 fonts to 1000.

So, to scale a font to render at a specific point size on a device with a specific resolution we take the resolution and point size to calculate the PPM value. Then we scale all the outline characters by multiplying by the PPM/UPM coefficient.

3.2 Coordinate Rounding, gridfitting

The final output is on a discrete raster, so the scaled outline coordinates need to be rounded somehow to integer values.

For example, if you have an outline point with coordinates (120, 100) and scale it down 7 times, you will get the coordinates (17.1429, 14.2857). After rounding to the closest integer values, the resulting coordinates will be (17, 14) and so the rounding error will be 0.1429 (0.84%) pixels for the horizontal coordinate and 0.2857 (2%) for the vertical coordinate. If we instead scale this point down 13 times, then the scaling errors will be 2.5% for the vertical coordinate and 3.8% for the horizontal coordinate. You can see that the rounding error increases as the size of the final outline is reduced.

To minimize rounding errors font rasterizers use special algorithms that slightly change scaled outlines to get better results on devices with low and medium

resolution. This process is called gridfitting. Algorithms that gridfit outlines use additional information stored with an outline's definition. These instructions are referred to as hints. Hints usually define the most important proportions of characters, the positions of critical elements of characters, and a set of rules for outline modification.

For perfect-looking fonts it's not enough to define the characters' outlines, you must also provide hints. The process of specifying the hints is quaintly called hinting.

3.3 TrueType and Type 1 hints

The two most commonly used font formats (Type 1 and TrueType) use very different hinting instructions and it is not always possible to automatically convert Type 1 hints to TrueType hints.

In Type 1 fonts, hints declare the most important dimensions in the characters, like the position and width of the crossbar of the letter 'H.'

Font hints in TrueType (called instructions) usually directly control the movement of points and the rounding of point coordinates. A TrueType hinting program is written in a special low-level programming language, compiled into the TrueType glyph description, preprogram or font program, specifying a particular action to perform not just on the current outline, but also on other data structures, or on the program's own flow of control. TrueType instructions include looping, conditional branching and function calls, arithmetic and logical operations. This makes TrueType hinting very flexible and powerful but also too complex to program directly. Usually, a smaller set of higher-level instructions are used to define hints. These instructions are compiled to native TrueType hinting language during font export.

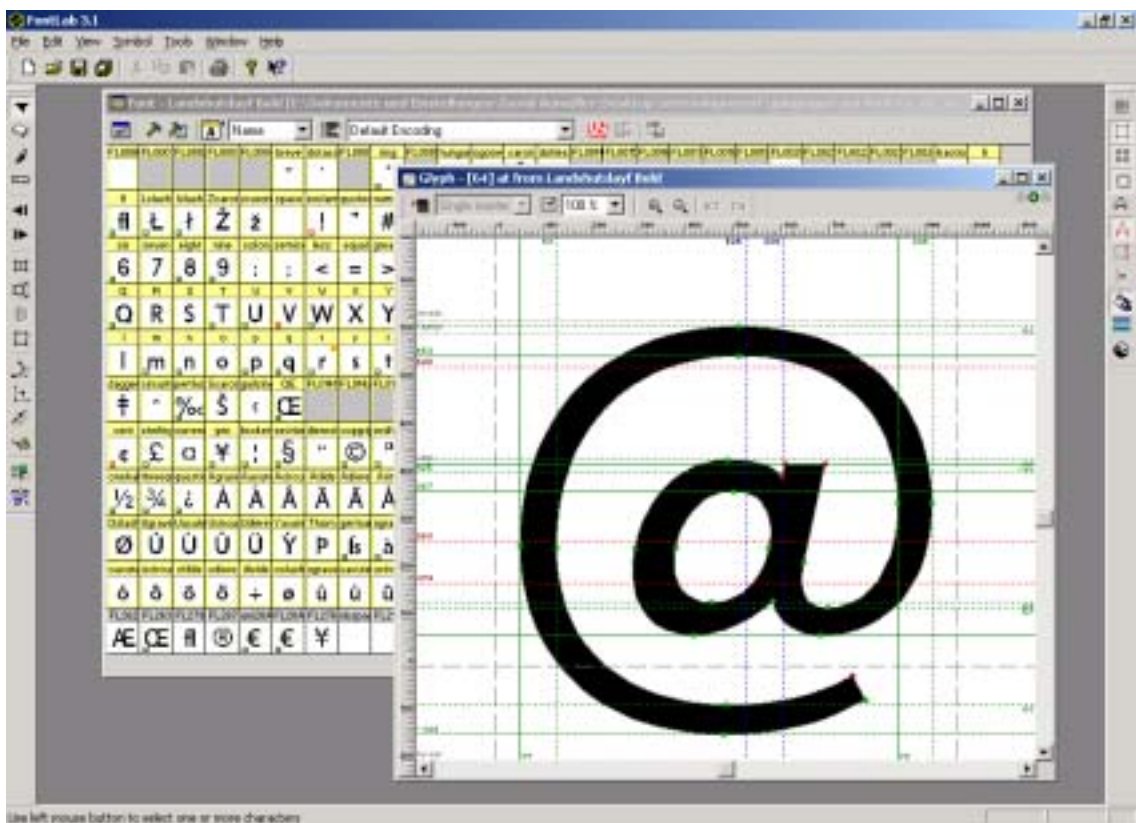
3.3.1 Auto-hinting

The algorithms used are generally not known to the user, so best results are achieved via trial and error. Algorithms typically analyse the features of *glyphs, and attempt to hint similar features the same way. While it is unfeasible that a computer should make the same hinting decisions as a trained typographer, auto-hinting on simple text fonts nearly always improves the font from an unhinted version. However with complex fonts, or with logos, there may well be no overall improvement. An expert typographic engineer can always substantially improve on an auto-hinted font.

3.3.2 Delta-hinting

Deltas are special instructions in a TrueType font which nudge the control points

of the glyph outline at particular ppem (pixels per em) sizes. Using deltas, you can usually define the exact bitmap you require at crucial small sizes. (Very tedious work.) Deltas are of the form: "At 12 ppem move control point 5 by 1.25 pixels". At Galápagos Design Group with Mike Leary being the hinting-expert, Delta-hinting was invented. The designers use a proprietary tool called "butt" which even allows the designer to turn on / turn off singular pixels in the target size (the software adjusts the outline to this size accordingly) and not just drag point by point of the outline definitions.



My own font, designed in FontLab

4 OpenType

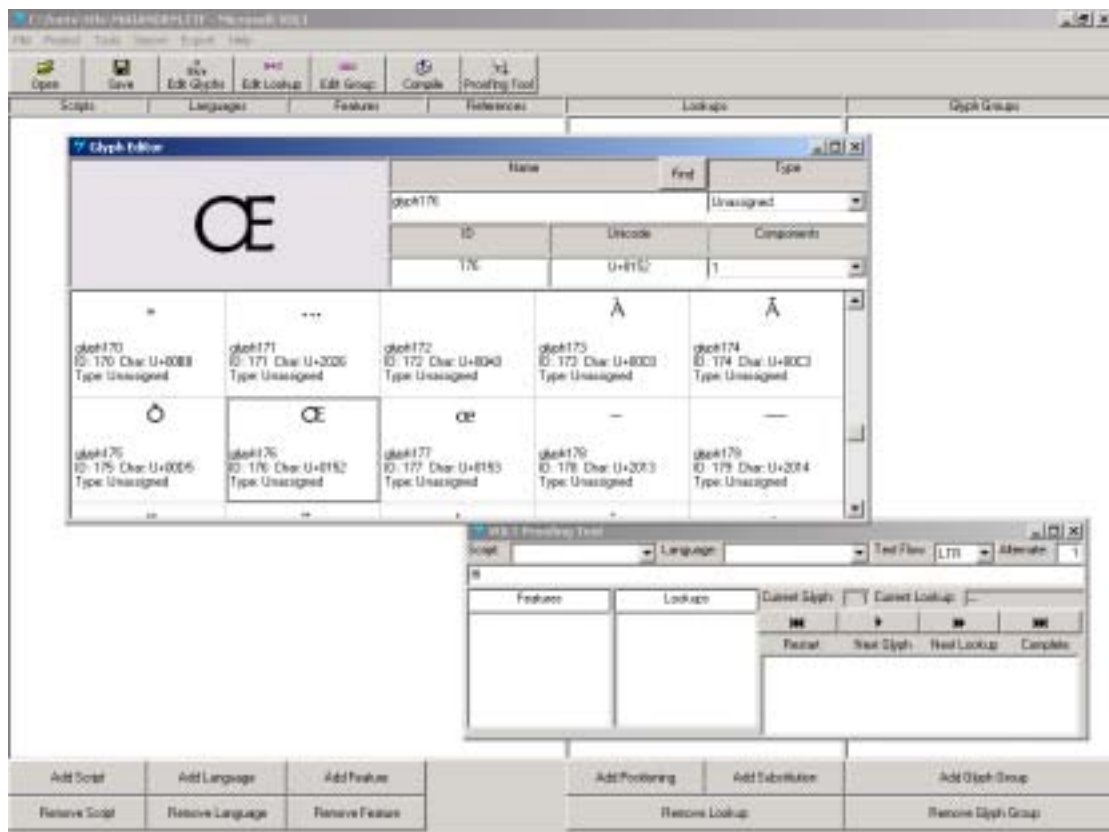
4.1 OpenType Layout

One of the OpenType format's most important features is its ability to support increased typographic complexity within a single font; from multiple styles of figures to complex script-specific ligatures, OpenType allows for typographic richness which was previously quite difficult to achieve. OpenType fonts do this by containing extra data which can be accessed by OpenType-aware operating systems and applications. While OpenType fonts are backward-compatible with pre-OpenType programs (OpenType supersedes both TrueType and Adobe Type 1 formats), the new layout features will not work with older programs. With Adobe's page-layout package InDesign, typographers and designers have access to a program which supports a selection of OpenType Layout (OTL) features. Though there have been a few OpenType fonts released by Microsoft and Adobe, the vast majority of existing fonts can't benefit from InDesign's OTL support in their current state. This is where applications like Microsoft's Visual OpenType Layout Tool come in: VOLT allows a font designer or developer to take an existing font and add OTL support, thereby allowing the font users to take advantage of the OpenType features in programs like InDesign.

However, the current release (version 1.5) of InDesign supports just a few of these features. Fortunately, the ones supported are some of the most useful to those attempting to produce high-quality, Latin script-based documents and publications. The seven features InDesign supports are:

1. `smcp`: allows for the use of true small caps, not spindly, computer-scaled caps which produce bad typographic color
2. `liga`: places ligatures (such as 'fi') while retaining the original, unaltered text
3. `kern`: allows for class-based and more complex sorts of kerning (in addition to the font's KERN table data)
4. `onum`: substitutes lining numerals for old style (lower-case) numerals
5. `pnum`: substitutes tabular numerals for proportional width numerals
6. `case`: provides support for case-sensitive substitutions (such as raised brackets for an all-caps settings)
7. `aalt`: this is the 'access all alternates' feature and it is implemented within InDesign's 'Insert/Replace Character' tool.

Every OpenType font doesn't need to support all of these features (and it may indeed support others not covered here); of the seven features InDesign supports, we'll focus on the four we think are the most generally useful: liga, smcp, onum, and case. Also, it is important to understand that our method of implementing these features is not the only way of going about the job; different fonts are created with different uses in mind — this is meant solely as a basic introduction to adding OTL features with VOLT.



VOLT – Visual OpenType Layout Tool

5 Content Management

5.1 Some rules to manage content

1. Form (layout and design) must be separated from content (primarily text). Although the words and pictures of a print layout are related, they don't necessarily work together the same way on the Web or on a CD-ROM as they do in print. The media on which content will be used should be taken into consideration from the beginning, not done as an afterthought. At the very least, layouts must change, and production issues such as the more limited color palettes and image resolution on the Web must be considered. Separation form and content also provides flexibility in terms of where each is used.
2. Content should no longer live in a nicely nested series of folders on several computers. It must reside in one or more data repositories (databases) from which it can be drawn for multiple uses. Text and graphics can be tagged to facilitate retrieval but still remain independent of each other.
3. Automation is essential. Re-purposing becomes increasingly labor intensive as the volume of material increases.
4. Content must remain media-neutral. This gives it the flexibility to be used in many different ways. For example, let's think about the screen of a cell phone or PDA.

5.2 HTML versus XML

Interwoven in all this is XML (extended markup language). Because it is media-neutral, content in XML format can be sent to many different devices for viewing, and sometimes printing (see my example below). XML must be transformed into other formats along the way, but its neutrality gives it the flexibility necessary to be a good framework for re-purposing data.

HTML (hypertext markup language) still has its uses, and it is the root of XML. The idea of creating tags, the structure of tags and many of the tags themselves are common to both schemas. But HTML only allows you to create static pages for viewing on a Web browser.

XML is a system that lets you make up your own tags, rather than depending only on those defined by the W3C (www.w3c.org). XML lets you define any type of form. It allows you to separate content – what you have – from presentation, which is how it looks. This lets you write separate presentation systems, or

templates, for interactive TV, broadband, PDAs and wireless devices. The device that accesses your site will be directed to the correct template. XML also lets you change the look and feel of your Web site quickly – you don't have to edit every page. Change the template, and the entire site's appearance changes. Most important, however, XML templates are the perfect interface for a database. All great sites today are based on databases because most computer data consists of databases.

Some browser nowadays can display XML, using Cascading Style Sheets (CSS), which defines the layout for the different XML-tags.

5.3 Perl scripts by myself

5.3.1 Generating a whole website

Using the Perl script I wrote, the webmaster just needs to set up a template html-file and write the content into another file (see `content_example.html` below; the source is conform to XML). The script merges the two files and generates for each `<FILE>` statement in the content-file a unique html-file; it also automatically generates the navigation-menu on the left side (which changes depending on your location in the 'tree'). I also used this technique to build the site for the practical course, the students' project called "durchblick⁸".

`content_example.html`

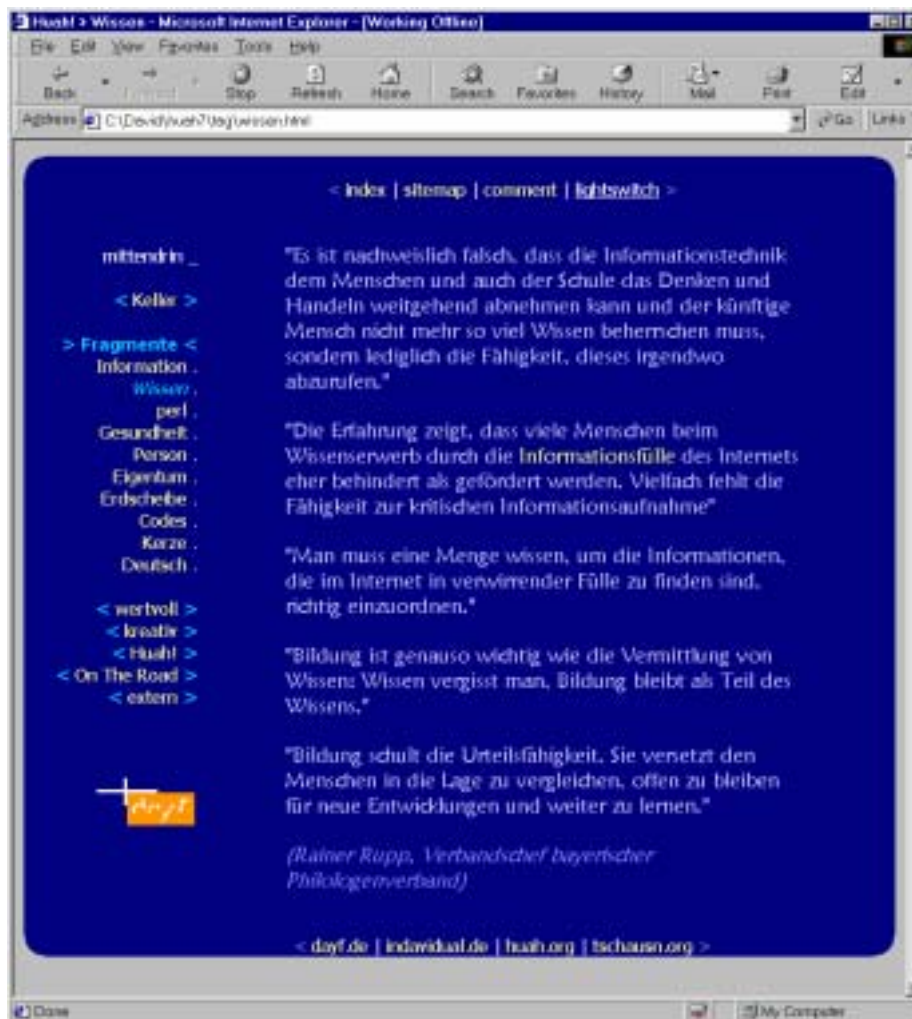
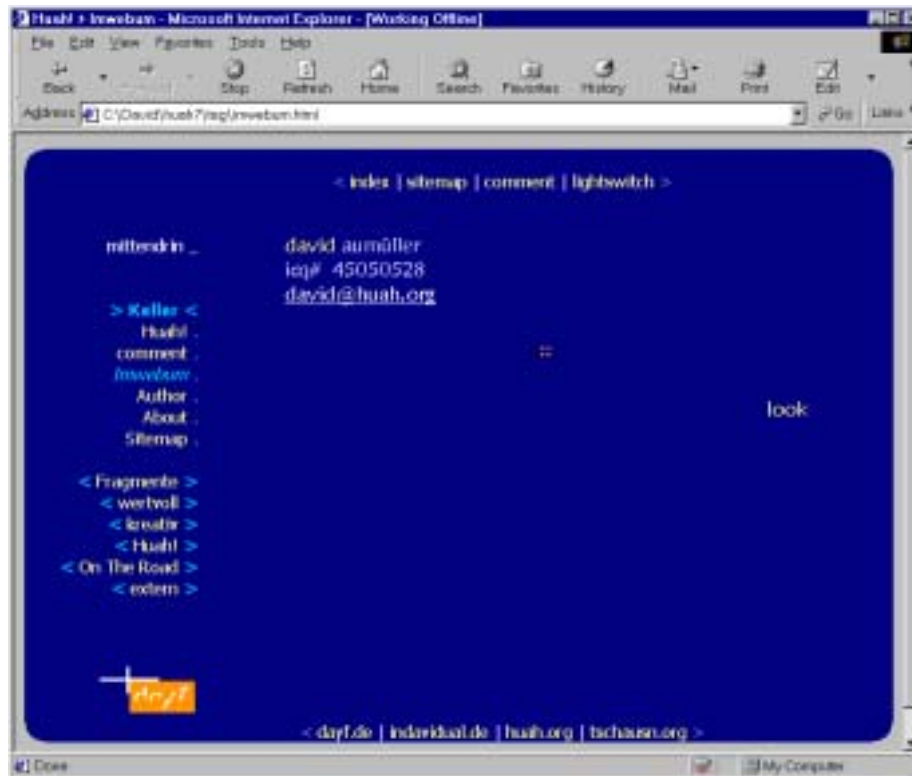
```
<FILE name="imwebum.html" title="Imwebum" category="Keller" menu="yes" date="20010101">
  <p><a href="ich.html">david</a> aumüller<br>
<a href="mailto:david@huah.org">david@huah.org</a></p>
  <p align="right"><a href="david.html">look</a></p>
</FILE>

<FILE name="worte.html" title="Worte" category="Fragmente" menu="no" date="20010117">
  <p align="right">"worte, worte, worte..."<br><i>(hamlet)</i></p>
</FILE>

<FILE name="erde.html" title="Erdscheib" category="Fragmente" menu="yes"
date="20010101">
  <p align="right">"Die Wissenschaft hat zweifelsfrei festgestellt,
dass die Erde eine Scheibe sei."<br><i>(die goldenen zitronen)</i></p>
</FILE>

<FILE name="photographs.html" title="photographs" category="On The Road" menu="yes"
date="20010201">
  <p>some pictures I took with the Kodak Digital Camera 215 ...</p>
  <ul><li>x-mas and new year's eve in
<a href=http://www.sendpix.com/albums/010110/09531b00e923c00013007ef
target="_blank">miami</a></li>
  </ul><p>the old selection aka
<a href=http://www.hdm-stuttgart.de/~da03/schublade/page_01.html
target="_blank">asservatenkammer</a> from dayf.de</p>
</FILE>
```

⁸ www.hdm-stuttgart.de/durchblick



Two pages from a well-structured website.

5.3.2 XML to PDF transformation

This script takes the XML source, which in the shown example is the data of a website-guest-book, processes it and generates a nice PDF file using the public domain PDFLIB for Perl (from pdf.lib.com, of course also available for PHP and other programming languages), i. e. Adobe Acrobat is not needed to generate the PDF! Though, in the current stage of this library it is pretty tough to format the output. You got to come up with your own algorithm for breaking paragraphs into lines; every line has to be put on the page via x- and y-coordinates in DTP-points. But considering to put the script online onto a web-server, the PDF could be generated each time a new entry into the guest-book is made. Another field of this technique might be an online generation of a customer's bill or whatever. See Appendix C.

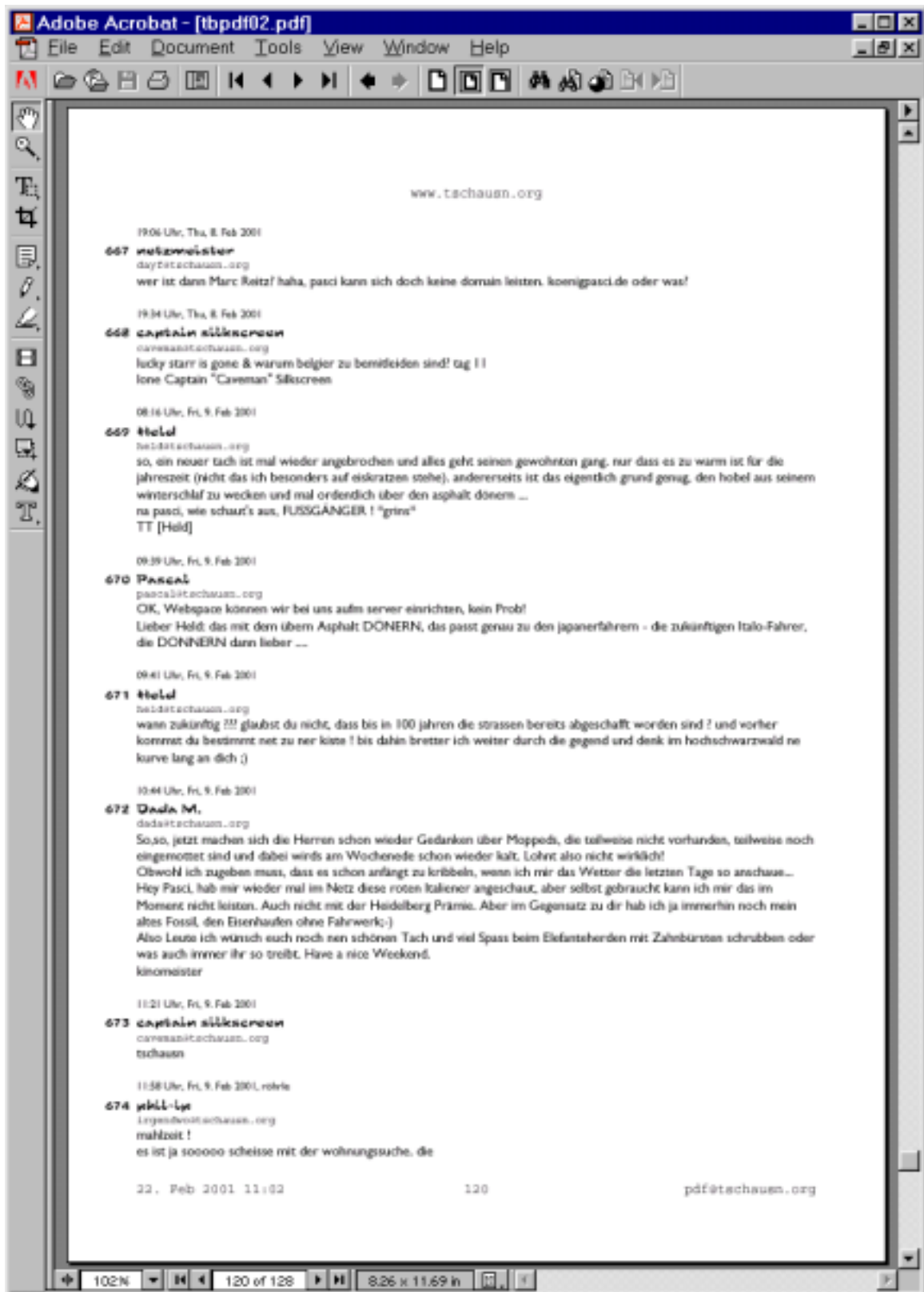
source.xml

```
<EINTRAG>
  <DATUM stunde="19" minute="06" wotag="Thu" tag="8" monat="Feb" jahr="2001"/>
  <NAME>netzmeister</NAME>
  <EMAIL>dayf@tschausn.org</EMAIL>
  <URL>http://</URL>
  <ORT>tschausnhausn </ORT>
  <COMMENT>wer ist dann Marc Reitz? haha, pasci kann sich doch keine domain leisten.
  koenigpasci.de oder was?</COMMENT>
  <LOGS>63.214.70.184 dialup-63.214.70.184.boston1.level3.net Mozilla/4.0 (compatible;
  MSIE 5.0; Windows 98) Opera 5.02 [en]</LOGS>
</EINTRAG>

<EINTRAG>
  <DATUM stunde="19" minute="34" wotag="Thu" tag="8" monat="Feb" jahr="2001"/>
  <NAME>captain silkscreen</NAME>
  <EMAIL>caveman@tschausn.org</EMAIL>
  <URL>http://</URL>
  <ORT>tschausnhausn </ORT>
  <COMMENT>lucky starr is gone & warum belgier zu bemitleiden sind? tag
  11<br><br><br><br>lone Captain "Caveman" Silkscreen</COMMENT>
</EINTRAG>

<EINTRAG>
  <DATUM stunde="08" minute="16" wotag="Fri" tag="9" monat="Feb" jahr="2001"/>
  <NAME>Held</NAME>
  <EMAIL>held@tschausn.org</EMAIL>
  <URL>http://</URL>
  <ORT>tschausnhausn </ORT>
  <COMMENT>so, ein neuer tach ist mal wieder angebrochen und alles geht seinen gewohnten
  gang. nur dass es zu warm ist für die jahreszeit (nicht das ich besonders auf eiskratzen
  stehe). andererseits ist das eigentlich grund genug, den hobel aus seinem winterschlaf
  zu wecken und mal ordentlich über den asphalt dönern ... <br>na pasci, wie schaut's aus,
  FUSSGÄNGER ! *grins*<br><br>TT [Held]</COMMENT>
</EINTRAG>

<EINTRAG>
  <DATUM stunde="09" minute="39" wotag="Fri" tag="9" monat="Feb" jahr="2001"/>
  <NAME>Pascal</NAME>
  <EMAIL>pascal@tschausn.org</EMAIL>
  <URL>http://</URL>
  <ORT>tschausnhausn </ORT>
  <COMMENT>OK, Webspace können wir bei uns aufm server einrichten, kein Prob!<br>Lieber
  Held: das mit dem übern Asphalt DÖNERN, das passt genau zu den japanerfahrern - die
  zukünftigen Italo-Fahrer, die DONNERN dann lieber ....</COMMENT>
  <LOGS>62.104.212.91 ffm2-t6-2.mcbone.net Mozilla/4.0 (compatible; MSIE 4.5;
  Mac_PowerPC)</LOGS>
</EINTRAG>
```



One of over hundred pages from the PDF document rendered by the Perl script.

6 Result

Looking back at these six months in Littleton, Massachusetts I feel contented. It was not the easiest step to leave behind all the comfort but very interesting to go to an unfamiliar place, settle down for a while, get to know charming people and of course summon new knowledge. Knowledge of type design – I got a much better eye for shapes now – and the technology behind fonts, as well as foreign culture and the English language.

I became proficient with font development tools and related software applications. I practiced the hinting of fonts and post production processes, quality assurance testing on industry standard platforms, including SUN, Macintosh and PC. I learned to troubleshoot problems between font technologies and applications. Automation of recurring tasks was another specialty of my working fields. With my knowledge of web design I was able to eliminate browser compatibility problems on various websites and perfection scripts on the server side.



There might be better locations for an internship than Littleton in Massachusetts, which is not exaggerated situated “in the middle of nowhere”. On the other hand I must be glad that there was a train station not too far off. So the commuter rail

brought me back and forth to Boston every week-end. The evenings during the week I enjoyed the silence in the wooden house where I was living as a tenant with a language teacher.

7 Appendix

- A Getting familiar with the tools, curve-handling (cubic splines, beziers)
- B Building a font from scratch, step by step as outlined above
- C PDF Document rendered by PDFLIB with Perl, as well as by Donald E. Knuth's LaTeX
- D WAD to RR – A letter about designing type
The Museum of Printing

8 Bibliography

1. Marvin Bryan, Digital Typography Sourcebook, Wiley Computer Publishing, 1997, John Wiley & Sons, Inc
2. Walter Tracy, Letters of Credit, 1986 by David. R. Godine, Publisher, Inc.
3. Noel Ward, Electronic Publishing, December 2000, PennWell Corporation, page 38
4. Dana Blankenhorn, Publish, February 2001, Publish Media, page 21
5. FontLab – Power Typography, The Advanced Font Editor, 1999 Pyrus North America, Ltd.
6. Robert Bringhurst, The elements of typographic style, Vancouver BC Canada: Hartley & Marks, Publishers, 1992, 1996, 1997
7. Microsoft Corporation, The Microsoft OpenType Specification. Redmond, Washington. Microsoft Corporation. 1990-1998
8. Microsoft Corporation, <http://www.microsoft.com/typography/tt/tt.htm>
9. Adobe Corporation, Adobe OpenType User Guide, <http://www.adobe.com>
10. Friedrich Friedl, Nicolaus Ott, Bernhard Stein, Typography – When Who How, Könemann Verlagsgesellschaft mbH, Köln 1998