

Project Runeberg's Electronic Facsimile Editions of Nordic Literature

Lars Aronsson

Linköping University, Sweden
lars@aronsson.se

Abstract

Project Runeberg has been publishing Nordic literature on the Internet since 1992. The project is based at Linköping University, Sweden. In the fall of 1998 a series of experiments were conducted to shift the project from e-text to facsimile images of text pages. The benefits and drawbacks of this shift are evaluated in this paper, and a cost model is presented together with some design decisions and implementation details.

Introduction

This paper describes the latest development in Project Runeberg, its new technology for *electronic facsimile editions*. The underlying technical concepts and components are neither novel nor very advanced. The focus of this text is to demonstrate step-by-step how these concepts can be combined into a cost-effective, predictable, scaleable, and user-friendly system for online presentation of paper documents.

In our case, this technology is demonstrated on 19th century printed literature from Sweden and the Nordic countries. But the same methods could just as well be applied to printed sheet music, to administrative documents in a company intranet, or manuscript collections. This work was funded by Linköping University [1].

Much of the inspiration for this work was contributed by the University of Michigan's *Making of America* (MOA) project [3].

The next section provides a background to Project Runeberg, which is our operational environment.

Background

Project Runeberg [5] publishes free electronic editions of Nordic literature on the Internet. This started in December 1992 as an activity within Lysator [2], a students' computer club at Linköping University [1] in Linköping, Sweden. During the first years, a group of students developed the project in their unpaid spare time. A network of volunteers was built among remote users of the growing electronic text collection. Being computer science students and pioneering Internet users, they got inspiration from the development of free software, such as the Linux operating system, and of course from Project Gutenberg [4].

The project's name brings associations both to Project Gutenberg and to Finland's great 19th century poet Johan Ludvig Runeberg. Finland, having been a part of the Swedish kingdom for 600 years, was ceded to Russia in the war of 1808–1809. Johan Ludvig Runeberg's poem *Fänrik Ståls sägner*, 1848–1860, recalled the memory of this war in a way that helped to prepare the ground for Finland's declaration of independence in 1917. This poem was the first work of Nordic literature published by Project Runeberg in early 1993.

Project Runeberg has since received some funding from the university, including the work leading to this paper, but also continued to rely heavily on voluntary efforts, as this improves productivity, moral, and legitimacy. Readers who are dissatisfied with the state of the collections are encouraged to help improve them, rather than complain. Taxpayers who want their money's worth are informed that most of the work was voluntary contributions.

For the first five years, only electronic *text* editions were produced. These are made available by WWW technology (HTTP/HTML). Initially, plain text files were served by Gopher and FTP, but the entire collections have been converted to HTML, starting in early 1994. A strict subset of manually edited HTML has been used, and the files still look good in a text editor. Readers are encouraged to 'view source' in their web browsers.

A well defined file directory structure was outlined when the collections were converted to HTML, resulting in a scaleable and manageable file tree and persistent, bookmarkable, and printable URLs without the need for external services or standards. Long filenames, deep directory structures, and non-English letters in filenames are avoided.

One lesson learned from the early Gopher experience was that interactively served files should be in the range of 2–200 kilobytes. For plain text or HTML, this corresponds better to a chapter than a novel. Consequently, Project Runeberg serves each chapter of a book in an HTML file of its own. Chapters are linked by arrows pointing to the previous and next element in the sequence. For each book there is also a combined title page and table of contents that offers links to all chapters.

Another early lesson was that readers will not enter a Gopher or website by the main entrance, but will follow external links to any document within the structure. There is a need to welcome the incoming reader wherever she enters, to tell her where she is, and to make her feel at home. These requirements have led to the inclusion of a page header in every one of Project Runeberg's HTML documents, containing the project's logotype, the title of the current book, as well as the aforementioned arrows to the previous and next chapter.

Fortunately, Project Runeberg's page header was developed before frames were introduced in HTML. Frames did not offer any advantages to the existing solution, but rather the disadvantage of non-bookmarkable URLs, so Project Runeberg decided to do without frames. These page headers were also developed long before cascaded style sheets were introduced in HTML.

Duplicating the page header in every manually edited document would be impractical. A tempting solution would be to let a programmable web server add the page header "on the fly" as web pages are served. However, this approach has drawbacks in delivery speed, since computation is needed during delivery, as well as in reliability and testability, because more complex software is more likely to fail at a later point in time for reasons that could not be foreseen when the software was designed. Instead, a solution was chosen where two parallel file structures are maintained. The first file tree contains the source files, which are manually edited, and kept under a revision control system. These files do not contain the page header. A separate subdirectory is used for the files belonging to each book. After an update of these source files, the files for the book are copied from the source file tree to the corresponding position in a second file tree by a program that also adds the page header, and automatically inserts the right links to the previous and next file in the sequence. The web server presents a view of the second file tree. This method consumes more disk space, but storage is cheap and prices are expected to fall further. The solution has proven to be extremely reliable. After files have been copied, they can be inspected to verify the process. Any simple web server software can be used to present these ready-made HTML documents.

The important deliverables from the project are not the displayed web pages, but the manually edited source files, which are really plain text files in the ISO 8859-1 (latin-1) character set, with a minimum of added HTML markup. The automatically included page header not only contains the links mentioned above, but also color codes (black text with red links on white background), and some Dublin Core metadata. Some of the information needed for this is stored in databases, which also belong to the source file tree. Backup copies are taken of the source file tree, the databases, and the copying program, but not of the web file tree.

Problems

Three major problems have been found with Project Runeberg's electronic text editions of Nordic literature.

External criticism against Project Runeberg has been launched primarily on the basis of the inadequate selection and documentation of editions for digitization. Literature scholars need to know which edition they cite, and prefer to cite scholarly edited text critical editions or, in lack thereof, some well-known and respected edition, such as the author's "collected works" or a first edition.

Another known problem, and one with a less obvious solution, is the insufficient reliability of text transmission from paper to computer file. Whether a text edition is created by keyboard entry or scanner and OCR software (optical character recognition), text errors might result, no matter how careful the proofreading. Whenever this suspicion arises, the only solution is a visit to the nearest library, and this largely reduces the benefits of having the text available online.

Creation of electronic text editions requires manual proofreading, and higher needs for quality calls for more careful proofreading. This puts limits on productivity. If some texts are more carefully proofread than others, there is also no obvious way to declare the level of confidence. Extra investment in manual proofreading is not only costly, but the effects of the investment are also difficult or impossible to measure. This is the third known problem.

The first problem is easily solved by using recognized editions for digitization, and explicitly declare which edition was used. In Project Runeberg's more recent electronic editions, this information is provided in a "Preface to the Electronic Edition" in the web page that contains the combined title page and table of contents. This preface is often written in English, so to be understood by speakers of any or none of the Nordic languages.

The other two problems motivated the development of procedures for creating electronic *facsimile* editions as a complement to the *text* versions. These procedures are described next.

Procedures for Electronic Facsimile Editions

The previous section gave an introduction to Project Runeberg's electronic *text* editions that made up its collections until the summer of 1998. During the same fall, the following procedures were developed to create and present electronic *facsimile* editions.

Selection of Works, Editions, and Copies

Project Runeberg has set out to publish Nordic literature on the Internet. Since the start of the project in 1992 its definition of *Nordic* and *literature* has varied, with a trend towards a more liberal interpretation. If it were to be rewritten today, the charter might use the phrase "Nordic digital library".

Project Runeberg's collections are dominated by literature which is old enough for copyright

to have expired. Only in a few exceptional cases have authors granted permission for Project Runeberg to publish their works, and this is always the author's initiative. Project Runeberg spends no efforts on such initiatives. Works for digitization are selected by a random mix of intuition, user requests, and usage statistics from works already published.

Acceptable editions are those referenced in literature. A contemporary encyclopedia might provide important information. For some works, the first edition is less useful because the following editions contained substantial additions. For some works, parts that appeared in the first edition were excluded in later editions because of censorship. In these cases, the uncensored edition is very likely to be more interesting. Any abridged edition is likely to be the wrong choice. A posthumous edition of an author's complete collected works is generally a good choice.

The highly automated method for digitization described below implies destruction of books, feeding separate sheets through a scanner. This makes it necessary to purchase books from antiquarian booksellers. Not only the work and edition must be suitable for digitization, but also the copy. No parts must be missing, the copy must be bibliographically complete. When in doubt, the copy to be digitized is compared with other copies of the same edition that might be found in libraries or at antiquarian booksellers. References in literature and statements in the work itself might provide useful information.

Scanning

For the purpose of creating an electronic *facsimile* edition, the output from scanning should be image files. The option of scanning directly to an OCR program is not used here.

Affordable books with good or moderate paper quality can be disbound and scanned at a low cost with an automatic document feeder. Fold-out illustration plates are first removed for separate scanning in a flatbed scanner. Other books might require a flatbed scanner or a high resolution digital camera, with an increased labor cost. Scanning from microfilm might also be an option. Most printed items are best scanned in bitonal, i.e. black and white. Woodcuts and rastered photos require the scanner resolution to be higher than the raster resolution. Otherwise, moiré patterns will result. Project Runeberg uses a resolution of 600 dots per inch (dpi) for all bitonal scanning, which is the highest available resolution with most scanners, and generally produces good results. Bitonal images are stored in the TIFF format with G4 compression. The TIFF file format allows multiple pages to be stored in one file, and documentation or metadata to be added in TIFF headers, but these options are not used.

Items that cannot be represented in bitonal are currently scanned in 300 dpi color JPEG, but the volume of this kind is yet too small to make any definite conclusions.

One file is created for each scanned page. Filenames consisting of four digits are generated by the scanner software, starting at 0001 and counting upwards. A new subdirectory is used for each book volume, allowing for volumes of 9,999 pages, which is more than enough. Blank pages are scanned too, so odd numbered files always display the right half of an opening.

No consideration is given to printed pagination at this stage. In many cases, the procedure described here will result in an offset of two or four between filenames and printed page numbers. Non-paginated illustration plates make the offset increase further. This is compensated for by page indexing, described below.

Archiving Scanned Images on CD-ROM

Image files from several volumes and works are organized in a simple file directory structure stored on CD-ROM (CD-Recordable). In the most restrictive ISO 9660 level 1 filesystem, file and directory names are at most 8+3 characters, single-case A–Z and 0–9. The top level (root) directory contains only one item, a directory named RONEBERG. This directory contains a subdirectory for each digitized work. Appropriate subdirectory names are assigned centrally by Project Runeberg before scanning, and will constitute part of the work's URL when published on the web. Because the subdirectory names are centrally assigned and globally unique, they can be distributed over more or fewer disks depending on file sizes and available space. For single volume works, the subdirectory immediately contains the image files. For multivolume works, a further level of subdirectories is introduced, one directory for each volume.

No text files, metadata, or other documentation is stored on the CD-ROM, which is both the deliverable from the scanner operator and an archival medium. The directory structure is intended to be self-documenting by being as simple as possible. Further documentation is to be found in Project Runeberg's online databases and files, but not on the archival disks.

```
\runeberg                Top level
|
+-\rikskal                Single volume work
|  +- \0001.tif
|  +- \0002.tif
|  +-  ...
|
+-\sbh                    Multiple volumes
|  |
|  +- \a                    First volume
|  |  +- \0001.tif
|  |  +- \0002.tif
|  |  +-  ...
|  |
|  +- \b                    Second volume
|  |  +- \0001.tif
|  |  +- \0002.tif
|  |  +-  ...
...

```

Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is an automated procedure for extracting text from scanned images. OCR software is still improving, and prices are falling, with new versions.

Input to OCR is TIFF files from the archival CD-ROM, as described above. The output is a similar file structure, with one text file for each image file, having the same basename, but .TXT suffix. Only plain text output is used, even though new versions of OCR software support various output formats, including HTML. The character set is ISO 8859-1, which covers the Nordic languages.

Because the output .TXT files contain errors, and need manual proofreading, it is called *raw text*. The raw text files are typically 2 or 3 percent of the image file size, and do not need CD-ROM for storage.

Indexing Pages

As was mentioned in the section on scanning, printed page numbers easily get out of sync with numbered TIFF filenames. A translation is needed between the two. We have found it practical to provide a translation table in a simple text file, which we call "Pages.lst".

Like other simple database structures used in Project Runeberg, the filename extension .lst indicates a line-oriented text file with records separated by newlines and fields separated by the vertical bar (|). These files are never exposed to the web client, but are used by scripts and programs that produce HTML files in the web server's file tree. The hash sign (#) is a comment introducer in .lst files.

Records in Pages.lst contain two fields. The first is the basename of a page file, e.g. 0001, and the second field is the printable page number. The page sequence is defined by the order of lines in this file. A typical example would be

```
# mybook/Pages.lst
#
0001|title page
0002|reverse of title leaf
0003|i
0004|ii
0005|iii
0006|iv
0007|v
0008|(blank)
0009|1
0010|2
0011|3
0012|4
```

In this example, there is a title leaf and five pages with roman numerals before page number 1, resulting in an offset of eight between printed page numbers and filenames. This indexing method is perfectly flexible for any unusual way to paginate books, including duplication, omission, and backwards pagination.

If fold-out illustration plates were taken out to be scanned separately, they can easily be reinserted into the sequence by modifying the Pages.lst file, e.g.:

```
0349|341
0350|342
# non-paginated illustration plate for figure 1
fig1|figure 1
0351|343
0352|344
```

Indexing Articles

As described earlier, Project Runeberg's text editions present each chapter of a novel as an HTML file of its own. Poetry books are made up of poems, while journals and dictionaries have articles rather than chapters. The word *article* has been chosen as the common term for the parts of any electronic edition.

In addition to the page index described in the previous section, an article index is also required for each electronic facsimile edition. This too is a plain text file, named Articles.lst, and is the basis both for a table of contents and for a hypertext link structure, as described in the following section.

The complexity of the data structures that an article index must be able to represent is perhaps best understood if the example of a weekly magazine is considered. An article might span several pages, and one page can contain several articles. A burst of advertisement pages might appear in the middle of a long running article, and some articles will inevitably be "continued on page 72".

Records in Articles.lst have three fields. The second or middle field is the printable name of an article. The first field is the basename of an optional HTML file containing a text version of the article. This allows proofread text versions of articles to be provided as a value-added complement to the facsimile images. This first field is required only for the first record, where it must be "index", implying the full filename index.html, which represents the book's title page and table of contents. The third field is a list of filenames of pages displaying the contents belonging to the article, each one matching the first field of Pages.lst.

Considering the possible irregularities in pagination mentioned in the previous section, where the same page number might occur twice in the same volume, it is necessary to refer to pages by their unique filenames, and not by the numbers that might happen to be printed on some of the pages. The page filenames in the third field of Articles.lst are separated by a single whitespace.

As a shorthand for the most common case, simple page intervals can be constructed with the ASCII hyphen (-). These are not numeric intervals, but intervals in the sequence specified by the first field of the Pages.lst file. For the example in the previous section, the interval "0341-0343" would be equivalent to the enumeration "0341 0342 fig1 0343". The interval "0343-0341" would be invalid.

The following example shows a typical article index. The page having filename 0381, at the end of the volume, contains the printed table of contents. This is indicated by the last record of Articles.lst and also by the first record. The web version of index.html will contain an automatically generated table of contents based on this article index, and this is indeed the text version corresponding to the scanned image 0381.tif.

```
# mybook/Articles.lst
#
index|Title Page and Table of Contents|0001-0002 0381
|Preface|0003-0007
# Page 0008 is blank, remember
|Chapter 1. The Beginning|0009-0013
# Page 0013 contains an overlap of both chapters
|Chapter 2. The Continuation|0013-0055
# ...
|Chapter 17. The End|0378-0379
|Table of Contents|0381
# Pages 0380 and 0382 are blank, so not mentioned here
```

This step finalizes the creation of an electronic facsimile edition. The scanned images, the OCR output raw text files, the Pages.lst and Articles.lst index files, and a descriptive "Preface

to the Electronic Edition” in index.html make up the entire source code. The remaining creation of web pages is performed by a program that is the same for all such editions produced by Project Runeberg. There are no plans to release the source code for this program, but some principles for its operation are given in the next section.

Making a Web

Still in version 4, neither the Netscape Navigator nor Microsoft’s Internet Explorer contain native support for displaying TIFF G4 images. Plugins to this end are available from third parties, but require the user to go through a download and installation procedure which is both repelling and confusing. Users faced with the requirement to install new software are likely to leave the website or to ask technical support questions to the content provider (the editors of Project Runeberg). In some situations, such as Internet terminals in public libraries, users might not be allowed to install plugins or other software.

Project Runeberg has decided to copy the *Making of America* (MOA) [3] project’s concept of on-demand server-side conversion of scanned high resolution bitonal TIFF images, suitable for archiving and printing, to low resolution grayscale GIF images, suitable for online presentation. This image file format, although proprietary, is supported by all graphic web browsers. MOA even provides the C source code of a tif2gif conversion program, based on the libtiff function library, which supports the G4 compression mode.

Computer screens typically feature a resolution of 70–120 dpi, so the original 600 dpi image would appear enlarged by a factor of 5–8 if displayed without scaling. The tif2gif program prioritizes speed over quality and flexibility, which is the only realistic tradeoff for an on-demand application, and allows scaling by a limited set of integral fractions, 1:3, 1:4, 1:5, 1:6, and 1:8. Most print is readable in 120 dpi grayscale (1:5), but some small print requires 150 dpi (1:4). Only exceptional cases would require different resolutions. Project Runeberg’s current approach is to set a fixed scaling for each edition. The displayed resolution is documented in the ”Preface to the Electronic Edition” provided in the electronic edition’s title page.

Images are accessed through a CGI script that maintains a cache file directory of already converted GIF images. If the requested image is already available in GIF, it is delivered from the cache. Otherwise, the conversion program is started. The conversion happens in a matter of seconds, during which the end user waits for the image to download. The cache file directory is a tradeoff between conversion speed (on cache hit) and available online storage space. Currently, Project Runeberg semi-manually removes all cached GIF files that have not been accessed in 30 days. Statistics on the contents of the cache directory is important input to the proofreading process described below.

From a human interface perspective, interactively served world-wide web documents have an optimal size of 2–200 kilobytes. This corresponds to a single facsimile image. As a consequence of not using HTML frames, Project Runeberg produces an HTML file as a wrapper around each scanned page image. This file contains Project Runeberg’s standard page header and footer, which provide metadata as well as pointers to the previous and next page in the sequence defined by Pages.lst. The converted GIF is an inline image in this document. Below the scanned image, the raw OCR text is included inside a pair of <pre> </pre> tags. These HTML files are produced in the web file tree only, and are never seen in the source file tree.

External fulltext search engines such as AltaVista and Infoseek will find and index the raw OCR text. When a user gets a search hit on this page, she will first see the inline facsimile image of the book page. Only if she scrolls down, will she see the raw text.

In the source file tree, index.html contains a "Preface to the Electronic Edition". When copied to the web file tree, not only the page header and footer are added, but a table of contents is also automatically generated, based on Articles.lst. Printable page names are found in Pages.lst. If more files than index.html were mentioned in the first column of Articles.lst, they constitute a sequence of articles that is separate from the sequence of pages. Appropriate, bidirectional links between articles and pages are automatically inserted in the web versions of the HTML files.

Optional Proofreading

The initial electronic facsimile edition, as created by the procedures described above, while providing solutions to the reliability and productivity problems described in the background section, also has major drawbacks when compared to a text edition. The text version is downloaded faster, it can be copied-and-pasted to other applications, and it can be read on Braille terminals by blind or visually impaired users.

To improve the value of an electronic facsimile edition, the raw OCR text can be proofread and provided as a text version in parallel to the existing facsimile edition. In this, the facsimile images are immediately useful for remote proofreaders who don't have access to a paper copy of the digitized edition. They also serve as a reliability guarantee against any text errors that might remain after proofreading. One article at a time is proofread, and the resulting HTML file is mentioned in the first column of Articles.lst. When the web version of the edition is remade, new links will be generated in the sequence of articles and between the corresponding page wrappers and the new article.

The ability to make proofreading optional was a major motivator for introducing the facsimile technology. Proofreading is a manual effort, and a costly investment, and should be directed to where it is most needed. Project Runeberg has chosen an approach based on statistics and voluntary efforts.

Statistics from the web server's access log provides information on which individual facsimile images are accessed most often. Statistics based on the contents of the cache directory for converted facsimile images, also provide information on which pages are accessed more often than others.

In novels, the first chapter is likely to be most popular. In poetry books and dictionaries, a few articles (poems) might attract a significant portion of the interest. By only proofreading the most frequently accessed articles, the cost of manual labor can be kept at a minimum.

When an entire book page is available in a proofread text version, the raw text can be removed. When the web version of the edition is remade, only the inline scanned image will remain in the HTML wrapper. Fulltext search engines will find the better version of the text in the proofread HTML version.

Readers are also invited to volunteer as proofreaders for their favorite articles or books. This invitation is printed between the facsimile inline image and the raw text in the page wrapper HTML file, and is visible to anybody who scrolls down to see the raw text. Voluntary efforts have an economy of their own. An open and welcoming attitude from the content provider, with clear instructions for how the work is best made, will make some people very generous. Rather than doing other hobbies, many are happy to help build an online collection that they know will be available to all.

Numbers

In its first five and a half years (December 1992 to June 1998), Project Runeberg published just

over 200 electronic text editions of Nordic literature. The edition count is a poor measure, as one of them is the Bible with 5 megabytes of plain text, and others are small collections of just a few short poems. The text quality of these editions also vary significantly, as well as the methods that were used to create them.

The described procedures for electronic facsimile editions in Project Runeberg were developed during the fall of 1998. In this process, 40 new facsimile editions were produced, containing 20,000 book pages, resulting in 2 gigabytes of TIFF images. The residual contents of the cache directory of frequently accessed GIF images is some 500 megabytes. The digitized works represent a mix of different categories of literature, including poetry, novels, textbooks, a biographic dictionary, and a full year's issues of an illustrated magazine.

With the frames-free approach and HTML wrappers around each scanned image, Project Runeberg's total web page count has increased from 15,000 to 40,000. The HTML wrappers only use a few tens of megabytes, a fraction of the GIF image cache, which in turn is just a fraction of the storage needed for TIFF images. Storing TIFF images offline or in some slow "CD jukebox" mechanism is not an option for a web application.

Project Runeberg's web server has a weekly count of 30,000 to 40,000 interactive clicks, and an additional 80,000 web server accesses for inline images and search engine robots. No major increase in access count has yet been observed as a result of the new facsimile editions. Perhaps it is still too early for any such effects, but the statistics-based proofreading efforts have also helped keeping the web accesses to facsimile images low.

The production cost for these electronic facsimile editions has been found to be roughly proportional to the page count for most kinds of literature. The exception is dictionaries or similar works where there are more articles than pages. A simple measure is the number of articles per 100 pages, where novels and textbooks have less than 20, poetry books and magazines have between 20 and 200, but dictionaries have more than 200. We currently believe that adding an article, i.e. a line to the table of contents, is about as expensive as scanning an extra page. This would imply the production cost be proportional to the total line count of Pages.lst and Articles.lst.

However, more production experience is needed to analyze this relationship further.

Conclusions

Compared to electronic text editions or text critical editions, the creation of electronic facsimile editions contains a high degree of mechanical, non-intellectual tasks. This makes the production cost more predictable. The production speed is more predictable than using unpaid volunteers. Volunteers can still make a significant contribution in optional proofreading, but the initial usefulness of a facsimile edition is independent of their productivity.

The facsimile editions make more intensive use of processing power, storage space, and network bandwidth, and require less skilled human labor. This means costs will fall as technology develops.

These advantages would be easily be consumed if the resulting editions were less useful or reliable than ordinary electronic text. But text reliability is increased by presenting facsimile images of printed pages. The increased need for bandwidth is still a drawback for modem users, but this problem is believed to diminish with time.

Of the component technologies used in producing these facsimile editions, some are more mature than others. We believe that 600 dpi bitonal scanning and storage in the TIFF G4 file

format on CD-ROM will remain the best choice for very long. OCR quality is still improving with each software version, and we believe the ways we index pages and articles can be improved beyond what we have today. Even more dramatic improvements, however, can be made in the web presentation, for example in adding metadata and links between related subjects. We believe we have found a mix of work procedures where we invest most efforts where results are more permanent, while being flexible on higher levels.

References

- [1] Linköping University, <http://www.liu.se/>
- [2] Lysator, a students' computer club at Linköping University, early website, and the origin of Project Runeberg, <http://www.lysator.liu.se/>
- [3] Making of America (MOA), a presentation on the Internet of several thousands of American books from the period 1850–1870. Produced at the University of Michigan in cooperation with Cornell University, <http://www.umdl.umich.edu/moa/>
- [4] Project Gutenberg, <http://promo.net/pg/>
- [5] Project Runeberg, <http://www.lysator.liu.se/runeberg/>