PERSONALISING THE EDITORIAL MIX FOR A DIGITAL NEWSPAPER USING CONSTRAINT PROGRAMMING

MICHAEL LUNØE

Department of Informatics and Mathematical Modelling
Technical University of Denmark

August 2012

SUPERVISORS:
Michael Kai Petersen
Carsten Witt

[August 2012 – Technical University of Denmark]

ABSTRACT

Short summary of the contents...

CONTENTS

I EXPLORING THE FIELD 1

1 CREATING THE EDITORIAL MIX 3
- 1.1 What is the Editorial Mix 3
- 1.2 Personalising a Digital Newspaper 5
- 1.3 Contributions of Constraint Programming 6
- 1.4 Problem Description 7
  - 1.4.1 Personalisation Challenges 8
  - 1.4.2 Algorithmic Challenges 8

2 PERSPECTIVES AND INSPIRATIONS 9

3 FEATURES TO BE PERSONALISED 13
- 3.1 User Needs 13
- 3.2 Use Cases 16
- 3.3 Requirements 17
  - 3.3.1 Non-functional Requirements 17
  - 3.3.2 Functional Requirements 18
- 3.4 Delimitation 20

II USING WHAT WE HAVE LEARNT 23

4 PROPOSED DESIGN 25
- 4.1 Acquiring and Mining Data for Personalisation 25
  - 4.1.1 Implicit and Explicit Data Collection 25
  - 4.1.2 Word tagging 26
- 4.2 Interface 28
  - 4.2.1 Typeface 29
- 4.3 Constraints 29
  - 4.3.1 Layout Constraints 29
  - 4.3.2 Content Constraints 29

5 TECHNICAL SPECIFICATION 31
- 5.1 Server 31
- 5.2 Client Side 31

6 EVALUATION OF THE SOLUTION 35
  - 6.0.1 Initial Test 35
  - 6.0.2 Result 35
- 6.1 Test 35
  - 6.1.1 Layout 35
  - 6.1.2 Content 36
  - 6.1.3 Functionality 36
  - 6.1.4 Improvements 37

7 CONCLUSION 39

III APPENDIX 41

A USER NEEDS 43

A.1 Personas 43

A.1.1 Thomas: student medium formal education male of age 21 44

A.1.2 Laura: employed high formal education female of age 39 44

A.1.3 Marie: unemployed no or low formal education female of age 61 44

A.1.4 Carl: retired or other inactive high formal education male of age 69 45

A.2 Scenarios 45

A.2.1 Thomas 45

A.2.2 Laura 46

A.2.3 Marie 46

A.2.4 Carl 47

A.3 Business Case 47

A.3.1 Need 47

A.3.2 Approach 47

A.3.3 Benefit Per Cost 48

A.3.4 Competition 48

A.4 Requirements 48

A.4.1 Non-functional Requirements 48

A.4.2 Functional Requirements 49

BIBLIOGRAPHY 51

ACRONYMS

DRY Don't Repeat Yourself

API Application Programming Interface

UML Unified Modeling Language

Part I

EXPLORING THE FIELD

Text for part 1.

1 CREATING THE EDITORIAL MIX

This chapter introduces the editorial mix of a digital newspaper and which parameters to account for when composing the newspaper. It is afterwards discussed which of these parameters are suited for personalisation, and how this can be done. The proposed approach is then presented with its pros and cons, which results in a list of contributions this project has to the field of personalisation.

1.1 WHAT IS THE EDITORIAL MIX

In the conventional newspapers the editors job is to compose an intriguing front page that offers the contents of the sections that might interest the individual user. His challenge is to accommodate the needs of the newspapers segment of readers, divide the articles into sections, with a nice reading flow and attractive illustrations, and hand-pick articles to go on the front page. But what if a computer could do this?

[Perkowitz and Etzioni, 2000] decomposes the problem of synthesising adapted page into several subproblems[1] :

- What is the content (that is, set of items) of the index page?
- Does it have a coherent topic? What should its title be?
- How are the items on the page ordered?
- How are the items labeled?
- Is the page consistent with the site's overall graphical style?
- Is it appropriate to add the page to the site? If so, where?

[1] *In the subproblems stated here, "hyperlink" has been replaced by "item" to appreciate them more generally, rather than the original specific sense.*

Some efforts have been made to digitally calculate similarities between articles and based on a current article suggest similar reading material or use collaborate filtering to suggest articles based on other users reading behaviour. Some papers proposes a composition of articles from user picked RSS-feeds, which can, e.g. in the case of

Google Reader, be divided into sections. This comes close to conventional newspapers, but there is no ordering of the flow of articles. The ordering, flow and choice of relevant articles is here on referred to as the *relational* part of the editorial mix.

The solution for some digital newspapers are still to have an editor to create their coherent composed digital newspaper, like the New York Times or Wired Magazine. Flipboard, on the other hand, composes their editorial mix of articles from feeds and divides their pages into three or more rarely four or five articles[2] with excerpts and images, much like conventional newspapers front pages. How they choose their composition is kept a business secret, but it does seem to vary a lot, see Figure 1.

[2] *Flipboard includes specialised layouts with more articles per page for Twitter.*

Figure 1: A screenshot of composition of three articles in Flipboard, with different subjects, i.e. world crime, world finance and technology news.

Maybe put in a reference to a conventional newspaper.

It is hard to say if there is a control behind the placement of content other than the choice of featured and non-featured articles, but this is actually an example of a computationally composed newspaper. The placement and amount of room given for an article is here on referred to as the *spacial* part of the editorial mix.

Finally, subjects of articles have more relevance at some points in time than others and editors choose the amount of time stories should be available in, where RSS-readers just displays the newest articles first, which are not always the most relevant. This selection of articles within a chosen time frame is here on referred to as the *temporal* part of the editorial mix.

One thing that is vastly different from newspapers to RSS-readers is the ability to deliver personalised content, in that a user can choose which RSS-feeds to follow, whereas readers of newspapers need to navigate it in order to find interesting articles. Also where newspapers have quality assurance of its content, RSS-readers have a seemingly unlimited amount of articles.

1.2 PERSONALISING A DIGITAL NEWSPAPER

[Bush, 1945] describes a collective memory library machine that can be indexed, called the memex. Items in the library are linked together forming personal association trails. This is the early conception of the hypertext media that would later become the World Wide Web and later personalised web applications. With many respects that is what this project tries to achieve; i.e. link information in the form of articles together and present them in personalised trials defined by the user. As opposed to [Bush, 1945] proposed manual linking it is now possible to, e.g. classify and compute similarity automatically, which greatly aids the process.

User preferences are very diverse and it is therefore hard to accommodate every individual in a single solution. A digital solution must be bound to a specific domain, but must also be open for novel use.

"Web personalization is defined as any action that adapts the information or services provided by a Web site to the needs of a particular user or a set of users, taking advantage of the knowledge gained from the users' navigational behavior and individual interests, in combination with the content and the structure of the Web site."

– [Eirinaki and Vazirgiannis, 2003]

The three categories of the editorial mix can be described in the sense of personalisation as well. Accommodating user preference based on spacial personalisation is achieved by a placement of articles, temporal personalisation by selecting articles of higher news value based on their relevance time frame and finally, relational personalisation by selecting articles that provides more value based on their respective and collaborate topics. Temporal personalisation is also obtained by letting user preferences have a life time and decrease the preference influence on which articles to select as time passes.

All three categories are related as they each provide some value to the editorial mix, e.g. a different spacial placement of a specific article can provide a different composition of the editorial mix and therefore a different temporal and/or relational value to the user.

1.3 CONTRIBUTIONS OF CONSTRAINT PROGRAMMING

"Informally, declarative programming involves stating what *is computed, but not necessarily* how *it is computed. Equivalently, in the terminology of Kowalski's equation algorithm = logic + control, it involves stating the* logic *of an algorithm, but not necessarily the* control*."*

– [Lloyd, 1994]

As a declarative programming language, Constraint Programming (CP) offers means for describing the problem to be solved using constraints and a general purpose constraint solver. Once the general purpose solver is set up, the constraints can be defined to model the problem to be solved, but does not necessarily make it easy. However, the problem definition can easily be extended and/or modified afterwards.

"Ordinary people generally aren't interested (and rightly so) in low-level programming details – they just want to express the problem in some reasonably congenial way and let the system get on with solving the problem. [...] Having to deal only (or mostly) with the logic component simplifies many things for the programmer. First, (the logic component of) a declarative program is generally easier to write and to understand than a corresponding imperative program. Second, a declarative program is also easier to reason about and to transform, as much current research in functional and logic programming shows."

– [Lloyd, 1994]

Also, with logic comes precise solutions, but this does not come without a cost as stochastic approaches often beat logic by lengths. However, stochastic variables can be introduced to CP.

To be able to work with personalisation problems as Constraint Satisfaction Problems (CSPs) or Constraint Optimisation Problems (COPs), these need to be defined. The following descriptions of CSPs and COPs have been modified to fit personalisation problems from the original definitions provided by [Russell and Norvig, 2003] and [Apt, 2006].

A Constraint Satisfaction Problem is defined by the 3-tuple $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where $\mathcal{X}$ is the set of variables, $\mathcal{D}$ is the corresponding set of domains and $\mathcal{C}$ is the set of constraints on the variables.

Each variable has sub-domains corresponding to the sub-value of each value. And unlike [Russell and Norvig, 2003] and [Apt, 2006] a constraint is here defined as a function of specific variables returning a boolean value. Therefore the tuple of n variables with m sub-variables and k constraints on r and s number of variables, respectively, can be expanded to:

$$\left(\begin{array}{c} \mathcal{X} : \left\{ x_1 : \begin{pmatrix} x_1.a_1 \\ \dots \\ x_1.a_m \end{pmatrix}, \cdots, x_n : \begin{pmatrix} x_n.a_1 \\ \dots \\ x_n.a_m \end{pmatrix} \right\}, \\ \mathcal{D} : \left\{ d_1 : \begin{pmatrix} d_1.a_1 \\ \dots \\ d_1.a_m \end{pmatrix}, \cdots, d_n : \begin{pmatrix} d_n.a_1 \\ \dots \\ d_n.a_m \end{pmatrix} \right\}, \\ \mathcal{C} : \left\{ c_1 : func(x_i, \cdots, x_{i+r}) \to \mathbb{B}, \cdots, c_k : func(x_j, \cdots, x_{j+s}) \to \mathbb{B} \right\} \end{array} \right)$$

Where $\mathbb{B}$ is either `true` or `false`.

A CSP is a subset of a Constraint Optimisation Problem (COP) and a COP is defined by the 4-tuple $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{O})$, where the three first elements are defined as in a CSP and $\mathcal{O}$ is a set objective (or cost) functions on variables, that determines the quality of a current state. The set of objective functions can be described with the same structure as constraints in CSPs and can be expanded as follows with l number of functions on t and u number of variables, respectively.

$$\mathcal{O} : \left\{ o_1 : func(x_i, \cdots, x_{i+t}) \to \mathbb{R}, \cdots, o_l : func(x_j, \cdots, x_{j+u}) \to \mathbb{R} \right\}$$

Where $\mathbb{R}$ is the set of real numbers.

Satisfaction (or regular) constraints are also called hard constraints and objective functions, soft constraints because a solution can be found if all hard constraints are satisfied, whereas and optimal assignment is enough to satisfy objective functions.

1.4 PROBLEM DESCRIPTION

This project is a feasibility study of the implementation of CP in the field of personalisation. The criteria of success is whether it is possible to successfully implement the techniques of personalisation using CP and to make personalisation more accessible with the aid of CP. Therefore the project will be divided into two main areas; i.e. the an assessment of the use of CP in the context of personalisation and a direct application of this in the form of a personal digital newspaper, where CP is used to personalise the content and composition of a digital newspaper.

Many techniques for personalising digital solutions already exists, but the role of CP within this domain has not been determined. This project seeks to explore CP as a tool to make the personalisation of digital solutions more accessible.

1.4.1 *Personalisation Challenges*

In an attempt to introduce a personal editorial mix in the digital newspaper, the report will try to analyse which preferences the users will have with respect to the content, composition and the time frame of relevant articles. It will describe the search for articles to fit the user needs as an Constraint Optimisation Problem and try to solve it. What makes a newspaper is not only the accumulated content of its articles, but the arrangement of them. "Which articles should go where" is just as important, and the placement of articles in the newspaper should therefore go through an equal solving process.

1.4.2 *Algorithmic Challenges*

To be able to use and assess CP in the context of personalisation a full understanding must be acquired. Features that can be solved using CP will be modelled as a COP and solved. Furthermore, because the problem has a fixed budget for finding a solution, its algorithmic complexity will be analysed. The findings will be concluded in an evaluation of the applicability of CP to personalisation problems.

2 PERSPECTIVES AND INSPIRATIONS

Web personalisation is by [Mobasher, 2007] divided into phases of data collection and preprocessing, pattern discovery and evaluation, and applying the discovered knowledge in real-time to mediate between the user and the Web. There have been many suggestions on how to tackle these different processes of creating the interactive personalised digital newspaper. [de Buenaga Rodríguez *et al.*, 2004] proposes a strictly stochastic approach to dynamic personalisation obtained by characterisation of content and user's interests. Both implicit and explicit relevance feedback[1] is used to refine the user models. Stochastic approaches have the advantage of being effective, but often solves a very specific problem. Also, these approaches tend to get very complex in order to deliver promising results. Some cope with this by introducing logic to the problem like it is done in [Nilsson, 1999] with a spacial approach. In this project it is possible to benefit from the structure of the logic approach of CP and the effectiveness of a stochastic approach by introducing preference constraints with an objective function.

Reference for stochastic approaches?

Many uses the approach of computing the tf-idf similarity with a cosine function as the distance function, as it is done in [Díaz and Gervs, 2005] to apply relational personalisation. It is based on a set of keywords extracted from the news items and a set of training documents. This constitutes the initial approach for computing similarity in this project. However, [Abuzir and Vandamme, 2002] argues that semantic knowledge is more substantial than keywords.

Classification techniques can be applied in order to ease the task of selecting relevant articles and determine their mutual relationships. [de Buenaga Rodríguez *et al.*, 2004] uses a library of documents to train a categorisation algorithm and the users are then asked to select categories of which they have interest. [Abuzir and Vandamme, 2002], on the other hand uses a thesaurus of hierarchically, and to the task specifically, structured terms to index news articles. Results of the indexing are thereafter mapped with user

1 Implicit is when the (unaware) user's behaviour is recorded to determine relevance and explicit is where the user is aware of the action of giving the feedback.

profiles to select the relevant articles. In stead of using predefined root terms as the basis for a classification, WordNet can be used to obtain semantic knowledge for a document. WordNet is a large lexical database of English words and their relationships in the form of different graphs. [Bouras and Tsogkas, 2010] presents an algorithm for enriching articles using WordNet's hypernym-graphs. WordNet also contains similarity functions between words. These functions will later on constitute the next step for computing similarity in this project.

[Díaz and Gervs, 2005] does present the means of combining the use of categories and keywords, but this approach demands predefined categories, which must be kept updated in order to follow semantic changes to the field. The time limitations and prioritisation of this project did not allow for a thesaurus to be obtained to aid the classification and will therefore not be introduced to the solution. One, could also argue that semantic assumptions are made, when categories are predefined, which could lead to some false classification. Whereas the structure of [Bouras and Tsogkas, 2010]'s algorithm bases its semantic structure only on words from the article and the general semantic (and more neutral) structure that constitutes the basis for WordNet.

[Claypool *et al.*, 1999] presents a combination of content-based and collaborate filters to predict interest in articles based on a user profile.

[Claypool *et al.*, 1999] presents a front page design and available sections. A section can appear as their front page. relevance feedback

their editorial mix consists of ordering the articles by most predicted interest first.

[Díaz and Gervs, 2005] along with both a short- and long-term representation of the user models. Furthermore, a global user profile, to get the process of generating the user model started.

[Centeno *et al.*, 1999] incorporates temporal personalisation in that it is possible to ask for articles in the newspaper based on a specific period, but also by incorporate ageing of user interests.

[Esteban *et al.*, 2001] incorporates temporal features of their personalisation of a digital newspaper based on Yahoo! Spain. Automatic Categorisation of news items, long- and short-term user models.

In the explored literature users shows much interest in being able to turn pages as it is done in a regular newspaper. [Ihlström *et al.*, 2004] describes this as "open, turn pages, chose article, read and return".

[Díaz and Gervs, 2005] proposes the use of collaborate filtering to handle the problem of converging, which is what will happen if no non-personalised articles are introduced, but this still only concerns articles that are within the area of the users interest. If e.g. a user has not shown interest in politics, the news of Barack Obama becoming the President of USA will never be included in the newspaper. Instead a ratio between personalised and general articles will solve this issue, and since it is not within everyones interest to receive general news, this ratio should be adjustable.

Users navigate the newspaper using sections and headlines as the main entry points [Ihlström *et al.*, 2004] and these should therefore be kept in the digital version.

Users express that these should be put into menu [Ovesson and Wikström, 2005].

[Díaz and Gervs, 2005] proposes personalised excerpts from the articles to further ease the navigation.

There exist many different examples of preference modelling using CP, and [Abidi and Chong, 2004] describes a factual information system to find personal information, e.g. about healthcare. They present two constraints: (1) select only information-objects that correspond to the user-model and; (2) the content of the retained information-items do not contradict each other. This is an example of an editorial mix in that it incorporates the relational features, i.e. both between user and articles, and articles in between. However, they do not take into account the spacial part of their editorial mix, nor do they take into account any temporal features of the information needed. It is of cause notable that the user needs for a strictly factual information system are different than from a newspaper.

Another application of preference modelling using CP is [Vossen, 2005]. He proposes a CP approach to automatic playlist generation, which very much relates to what this project attempts to achieve. A playlist can be perceived as a, in this case, personal mix of songs. He presents constraints to exclude songs with certain attributes, to model that certain songs should be similar to each other or a user preference, to describe preference about the number of songs from a specific artist and the well-known `all-diff` constraint. These can be directly translated to the editorial mix of a newspaper, where the songs are articles and artists could be a specific author or content provider.

3 FEATURES TO BE PERSONALISED

This section analyses the user preferences and identifies which features should be modelled as personalisation constraints. It also seeks to define the default setting that consists the general model for a good editorial mix of a digital newspaper.

The focus of automatically generating the editorial mix introduces temporal, spacial and relational circumstances about the composition. But before these can be modelled as constraints it is necessary to look at the user needs of the application.

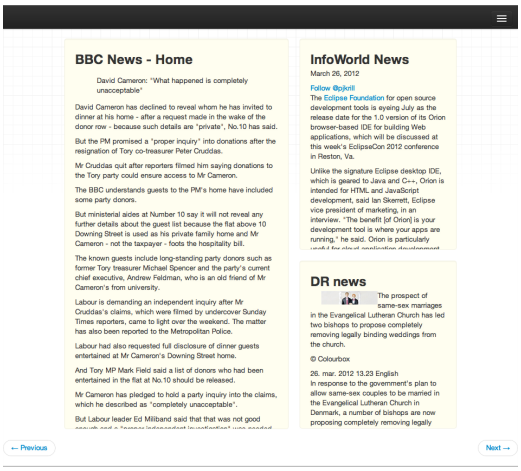
Some prerequisites must be stated in order to set the focus of the problem, but also for the potential user to relate to the product. The research done by [Ihlström *et al.*, 2004] and [Ovesson and Wikström, 2005] determines the preferable size of the digital newspaper to be 14.732 × 20.828cm ∼ size A5, which reflects the size of the iPad. Therefore this project will target the iPad as its primary device. The handheld device also introduces mobility, which is also of great preference to the potential users.

3.1 USER NEEDS

This section will define the user needs for the application. A full description of personas, scenarios and business case is found in appendix A on page 43.

After the definition of the initial requirements was done the first prototype was developed. Its main features followed the requirements on turning pages, choosing an article, read it and returning to the overview of articles, see Figure 2a.

After a preliminary small survey it became clear, that a column-based layout would be more attractive, and would provide a better opportunity to explore the editorial mix. With a column-based layout the digital newspaper would have more resemblance to conventional newspapers and therefore it was possible to apply some of the same principles of the editorial mix.



(a) Initial prototype layout with adjustable ratios between articles and a paged interface of each section.

(b) Second iteration of the prototype with an "endless" layout. Sections are placed beneath each other.

(c) Third iteration of the prototype with a column-based and "endless" layout. Sections are placed beneath each other.

(d) Third iteration of the prototype with a column-based and "endless" layout. Sections are placed beneath each other.

Figure 2: The figure shows three iterations of the prototype layout.

After the second prototype was developed it was tested. A specification of the test can be found in Table 1.

Table 1: Test Specification

Test subjects	The test was conducted on a total of 7 test subjects of ages between 21-29, and of different sex and occupation.
Participants	Each test was done with 1 test conductor and 1 test subject.
Materials	An iPad with the application running and a computer to write notes on the test subject's statements and propositions.
Description	The test subject was presented with the prototype layout seen in Figure 2c and 2d. The test was conducted as an informal qualitative talk with a basis in the test subject's interests in such a product. Transcripts from each test can be found at `lestrade.imm.dtu.dk/~ArtistRecommender/data/test-transcripts.zip`.

The prototype consisted of the basic navigation between subject categories, i.e. sections, and articles. Navigational choices was made in order to present the general idea of the framework, but where more crucial choices on its uses have not been made jet. This was also to encourage the test subjects to talk about what uses they would have of the presented framework. However, they were also asked about the navigational structure and indeed some changes had to be done.

The above presented preparatory work resulted in the following user needs:

- Read articles presented in a nice and digestible layout
- Get an easy overview of the content of the newspaper
- Easily navigate between articles with few touch-friendly interactions
- Read relevant articles based on user defined subjects

3.2 USE CASES

These user needs and the results from the conducted tests proposes some use cases from the application. Theses will presented here.

In order to fulfil the user needs more general use cases are derived. These and the web servers handling of them are shown in Figure 3.

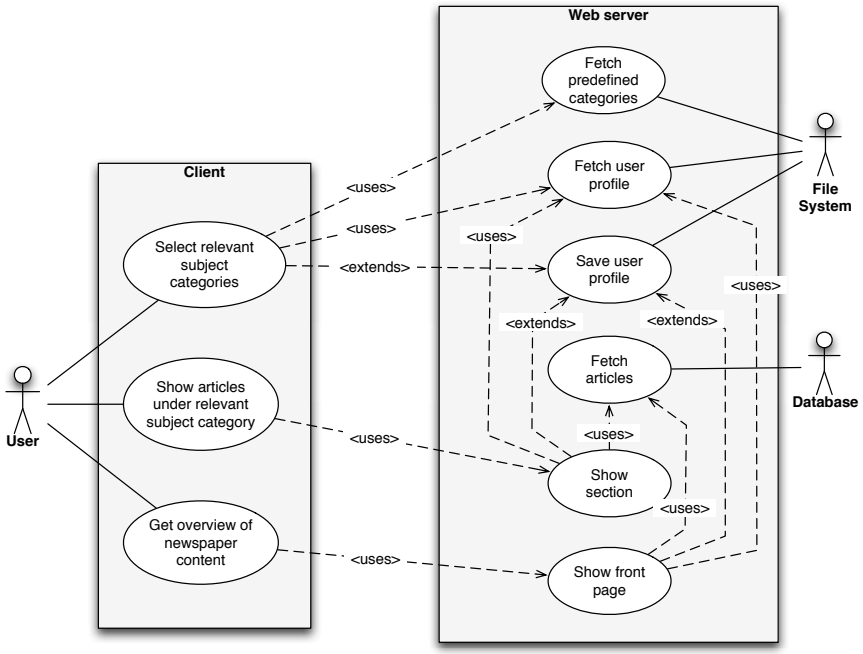


Figure 3: The figure shows the overall use cases of the system and how the web server acts to accomplish them.

The list of use cases shown in the figure are by no means a complete, but are selected because they have the closest relation to the user needs.

The user is able to select relevant subject categories to get an easy start with the application and the choices are thereafter saved to the user profile for later use. The user can select a subject category from the earlier chosen categories and browse articles from this subject within a section. If the user needs to get an overview of the content of the newspaper he can get the front page displayed, which contains a small collection of articles of most interest to the user.

In Figure 4 is shown a sequence diagram of what the system does in order to display the front page (or a section), when the user opens the application.

When the application is opened, or the application is changed to display a section, a background worker is initialised to compose its con-

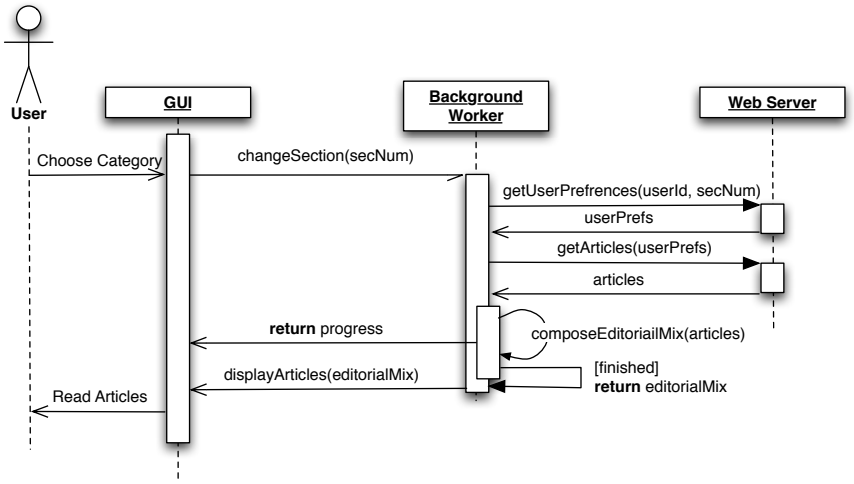


Figure 4: The figure shows a sequence diagram from when the user chooses a subject category until he can read articles from this subject. `secNum` is the section number to display (the front page is section 0), `userId` is a string that identifies the user, `userPrefs` is the user preferences on the specific section and `articles` is the library of articles to compose the editorial mix of.

taining mix of articles. Of cause, if the mix of articles in a section, or front page, have already been computed, it does not need to recompute it. The background worker needs to get both the user preferences of the chosen category and article that potentially fit the user preferences. While worker computes the editorial mix it sends messages to the user interface about the progress. When it finishes the user interface is asked to display the articles.

3.3 REQUIREMENTS

3.3.1 *Non-functional Requirements*

In the explored literature and the conducted tests users express some non-functional requirements. These are described in this section.

User needs states the requirement of having a clear overview of the content and as stated in [Ihlström *et al.*, 2004], this includes a clear marking of the beginning and the end of the articles and sections. This is obtained by both having a summery of the most interesting articles on the front page and by having a dropdown list of headlines in the top of section.

From the user needs it is also required that the system should be easily navigated and as stated by [Ovesson and Wikström, 2005], this should be through clickable sections, headlines and through paging. The layout, typography and design should be familiar to what is found in conventional newspapers, as stated by [Ihlström *et al.*, 2004]

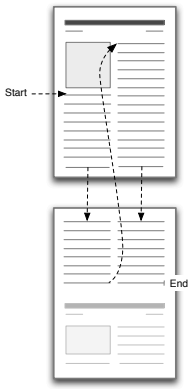


Figure 5: Reading pattern where the user has to scroll in order to see the full length of the column.

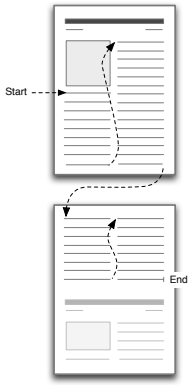


Figure 6: Reading pattern where the user can finish reading a whole page before scrolling to read the next.

and [Åkesson *et al.*, 2005]. This is achieved by choosing a structure that resembles that of a newspaper and displaying content in balanced columns. It should be possible for the user to read the newspaper screen by screen, meaning that trailing text should be put into a new set of columns whenever it exceeds the screen, see Figure 6 and 5.

- Ease of use
- both general and personal news (collaborate filtering solves that some news are not received, but are universally interesting [Díaz and Gervs, 2005])
- both images and videos - test
- a good ratio of graphical and textual - test
- front page should give a good overview of the content - test
- "news valuation, e.g. positioning of lead story" [Ihlström *et al.*, 2004, p. 7]
- mobility [Ihlström *et al.*, 2004, p. 7]
- continuous updates [Ihlström *et al.*, 2004, p. 7]
- "easy and intuitive navigation" [Ihlström *et al.*, 2004, p. 7]
- add video and sound [Ihlström *et al.*, 2004, p. 7]
- incorporate social community and social networks

3.3.2 *Functional Requirements*

Some technical requirements have also been gathered from the explored literature.

- "open, turn pages, chose article, read and return" [Ihlström *et al.*, 2004, p. 6]
- section headlines [Ovesson and Wikström, 2005, p. 6-7]
- article headlines
- article summaries / extracts [Díaz and Gervs, 2005]

- menu w. section headlines [Ovesson and Wikström, 2005, p. 8]
- page numbers [Ovesson and Wikström, 2005, p. 6-7]
- press "like" or key word based user profile (mark self or highlighted? right click to add): positive + negative list (keywords+categories [Abuzir and Vandamme, 2002], [Díaz and Gervs, 2005] and [de Buenaga Rodríguez *et al.*, 2004])
- full screen display of article
- organise into personalised sections
- opens in front page view (summery of newspaper 8 articles) [Ovesson and Wikström, 2005, p. 8]
- adjust variables
- share directly (grey out the ones who have read it)
- comment
- see friends comments
- "The presentation schema – headline, abstract, and text, together with a relevance value with respect to the user profile – rates the highest in terms of user satisfaction, and yet it is not the most frequent." [Diaz *et al.*, 2001]
- ability to search [Ihlström *et al.*, 2004, p. 7]
- Landscape + portrait [Ovesson and Wikström, 2005, p. 6-7]
- touch screen interaction [Ovesson and Wikström, 2005, p. 6-7]
- Functionality from online newspaper [Åkesson *et al.*, 2005]
- Name of columnist [de Buenaga Rodríguez *et al.*, 2004, p. 4]
- Transparency of implicit relevance feedback (see/modify current weights of categories) [de Buenaga Rodríguez *et al.*, 2004, p. 7]
- dynamic short-term + static long-term user profile [Abuzir and Vandamme, 2002], [Díaz and Gervs, 2005] and [de Buenaga Rodríguez *et al.*, 2004]

- relevance feedback [Abuzir and Vandamme, 2002], [Díaz and Gervs, 2005] and [de Buenaga Rodríguez *et al.*, 2004]

3.4 DELIMITATION

The proposed design in chapter 4 on page 25 should of cause account for requirements, but there are some requirements that will not be implemented due to prioritisation and time limits

The initial approach involved computing the tf-idf similarity between documents and the user and the documents in between using the Python libraries for this [Bird *et al.*, 2009]. This approach works on a bow (bag-of-words) with key words and weights representing a single item. The weight is computed by the number of occurrences in the provided text and a cosine distance determines the similarity. Python also provides an interface for working with WordNet – a large lexical database of English words and their relationships in the form of different graphs. This opens the door to a more in-depth analysis of the optained news items. [Bouras and Tsogkas, 2010] presents an algorithm for enriching articles using WordNet's hypernym-graphs. A hypernym graph is generated by the top 20% frequent keywords of an article and weighted by:

$$W(d,f) = 2 \cdot \frac{1}{1 + e^{-0.125(d^3 \frac{f}{TW})}} - 0.5$$

Where d stands for the node's depth in the graph (starting from root and moving downwards), f is the frequency of appearance of the node to the multiple graph paths and TW is the total number of words used to generate the hypernym graph.

In order to be able to work with hypernyms, the words must be converted to synsets. For each word there exists a synset for each use of the word, with the most frequently used first. Every synset is included at this point, but in a later stage this could be further focused by only using the top n. An analysis on how many percent of the words

Which design choices to focus on?

Introduce columns and remove pages to introduce newspaper like layout

Do not focus on summaries – out of scope,

Support images only, not video – can easily be introduced.

Not implement support for social networking, but included in the proposed design

Predefined categories is not a full list, but are of the most recurring in popular news sites.

In which period of time is an article relevant to a user? Maybe if it is still available, then it is still interesting – new approaches or discussion about the subject might arise. How do we control that a news item is not missed? Keep index of what has been viewed in addition to what has been read.

Part II

USING WHAT WE HAVE LEARNT

Text for part 2.

4 PROPOSED DESIGN

First and second iteration

4.1 ACQUIRING AND MINING DATA FOR PERSONALISATION

The [Dub, 2012] proposes 15 meta data elements, i.e. Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage and Rights.

categories predefined, but must be updated – later the recording of the user behaviours can supply these category choices. Also they should be based on the root terms presented in [Abuzir and Vandamme, 2002].

This section will describe the necessary data mining before it is possible to apply personalisation on the digital newspaper.

Won Kim, "Personalization: Definition, Status, and Challenges Ahead", Journal of Object Technology, Volume 1, no. 1 (May 2002), pp. 29-40

spacial, temporalt, topics (content) – en distribution over topics – (probabilistisk model – distribution af ord – referer til en lda model), lidt det samme at tale om et billede og musik.

Reference: [Blei, 2011]

4.1.1 *Implicit and Explicit Data Collection*

[Eirinaki and Vazirgiannis, 2003]

Introduce collaborate filtering

Introduce relevance feedback (click on article, time spend reading it and scroll)

Read "Personalization Techniques and Their Application.pdf"

using hash of url as ids.

looking for images and videos in the article.

4.1.2 *Word tagging*

tagging: maxent_treebank_pos_tagger, Treebank Part of Speech Tagger (Maximum entropy) – sikkert upenn corpus.

alternativt: Treebank Part of Speech Tagger (HHM)

wordnet_enrich: "W-kmeans: Clustering News Articles Using WordNet", 116262780379.pdf

wordnet sammensatte ord

citep "Text Classification Using WordNet Hypernyms.pdf" hypen density. This is better in 116262780379.pdf

using nouns and adjectives NOT nouns and verbs as in "Text Classification Using WordNet Hypernyms.pdf"

tagging på feeds er givet fra feedets navn. Kan bruges til at verificere min classification.

Opbygge en database med articler og bruge det som basis for similarity queries. wordnet_ic Information Content: Load an information content file from the wordnet_ic corpus.

[Shapira *et al.*, 2009] presents ontology levels from front page (overview) to a specific subject providing the ability to browse articles based on different levels delimitation. Also, that the most interesting articles should be placed on the front page and other relevant articles within each category. The same ability is provided through the rough overview on the front page, with the top articles of each section and the possibility to specify each section on the the preferred level of ontology. E.g. if the user wants articles from the broad category of business, this category can be chosen and through the time spend on diverse articles, the system will learn that this is the preference

Listing 1: cap

```
        >>> from nltk.corpus import wordnet_ic
        >>> brown_ic = wordnet_ic.ic('ic-brown.dat')
        >>> semcor_ic = wordnet_ic.ic('ic-semcor.dat')
        >>> dog.res_similarity(cat, brown_ic)
        7.9116665090365768
        >>> dog.res_similarity(cat, genesis_ic)
        7.1388833044805002


explain SELECT MAX(sim) as max_val FROM `hypens` WHERE hypen
    1 IN ( .... ) AND hypen2 IN ( ... ) GROUP BY hypen1;
+----+-------------+--------+-------+---------------+----------+---------+------+---------

| id | select_type | table  | type  | possible_keys | key
         | key_len | ref  | rows     | Extra       |
+----+-------------+--------+-------+---------------+----------+---------+------+---------

| 1  | SIMPLE      | hypens | index | hypen1_2      | hypen
    1_2 | 126     | NULL | 49472237 | Using where |
+----+-------------+--------+-------+---------------+----------+---------+------+---------
```

of the user. That is, of cause only given that the user acts his preference. Also, the Wordnet enriching of the articles works on the basis of ontology levels in the hierarchy of Wordnet structure.

It should as [Shapira *et al.*, 2009] proposes be possible to get the standard edition of the newspaper if the user does not need personalised news. This could be done by simply turning off the implicit recording of the user behaviour – this will handle some privacy issues as well.

4.2 INTERFACE

[Nielsen, 1997] on how users read on the web "They don't. People rarely read Web pages word by word; instead, they scan the page, picking out individual words and sentences."

"2. PDF Files for Online Reading: Users hate coming across a PDF file while browsing, because it breaks their flow. Even simple things like printing or saving documents are difficult because standard browser commands don't work. Layouts are often optimized for a sheet of paper, which rarely matches the size of the user's browser window. Bye-bye smooth scrolling. Hello tiny fonts. Worst of all, PDF is an undifferentiated blob of content that's hard to navigate."

"4. Non-Scannable Text: A wall of text is deadly for an interactive experience. Intimidating. Boring. Painful to read. Write for online, not print. To draw users into the text and support scannability, use well-documented tricks:

- subheads
- bulleted lists
- highlighted keywords
- short paragraphs
- the inverted pyramid
- a simple writing style, and
- de-fluffed language devoid of marketese.

"

"5. Fixed Font Size Respect the user's preferences and let them resize text as needed. Also, specify font sizes in relative terms — not as an absolute number of pixels." [Nielsen, 2011]

4.2.1 *Typeface*

Colour scheme: http://colorschemedesigner.com/#0042b1Tw0w0w0

Figure 7: Colour scheme of the interface [Stanicek, 2010]

colorschemedesigner.com citep

4.3 CONSTRAINTS

The constraints can be divided into groups: layout-/content-based, global/topic/neighbouring,

Technical argumentation for choices. Business case argumentation for choices (user needs). Reference to written articles about design choices – navigation: sections + headlines,

4.3.1 *Layout Constraints*

first focus: layout + one sections

3? columns images/no image base layout

4.3.2 *Content Constraints*

Content similarity relationships: article vs. neighbouring articles article vs. containing section/topic article vs. whole newspaper

breaking "[...] it was found that the best eight-item mix within an issue was not necessarily composed of the eight highest-readership items in that issue." [Haskins, 1965] (maximum audience coverage, which is no longer an issue due to personalisation)

$$\mathcal{C} = \left\{ \begin{array}{c} \texttt{all\_different}(a_i), \\ \texttt{sim}(a_i, a_{i+1}, 0.7, 0.9), \\ \texttt{sim}(a_i, a_{i-1}, 0.7, 0.9), \\ \texttt{sim}(a_i, a_j, 0.3, 0.9) tempo_i < tempo_{i+1} + 10 \wedge tempo_i < tempo_{i-1} + 10 \wedge \\ tempo_i > tempo_{i-1} - 10 \wedge tempo_i > tempo_{i+1} - 10, \end{array} \right\}$$

AIRussel p. 207 preference constraints. can often be encoded as costs on individual variable assignments. Solved either path-based or local. p. 216 Minimum-remaining-values, p. 217 Least-constraining-value.

Figure of the system design: Model (user model + meta data + constraints), View (layout + inteactions), Control (cp)

5 TECHNICAL SPECIFICATION

css: conditional styling

5.1 SERVER

5.2 CLIENT SIDE

worker.js to create a background worker to perform the constriant programming.

Model-View-Control using backbone.js

Paging: single page web apps + manipulation the browser history https://developer.mozilla.org/en/DOM/Manipulating_the_browser_history

Preference ordering of hard constraints or division between preference constraints and hard constraints.

Assignment from library in stead of arbitrary assignment? The latter is a more hypothetical approach. Providing the library as a constraint, where each variable assignment must have a unique combination from one of the possibilities of the constraint. (sim,breaking,chars,date,sections?,columns):list The former introduces an implicit constraint in that the general purpose solver can only choose from the library, thus can only choose a combination of values that exists.

Ranges can be optimised in space by converting them to integer ranges. This can be done by setting min = 0 and max = (b-a)/gap.

Furthermore each subdomain should be able to be represented by a set of ranges and atomic values. Propagating through values causes many iterations and a whole range may be discarded by looking at its maximum and minimum value. However, if the range holds a potential valid value (solution to a variable) it can be divided into

smaller ranges and their minimum and maximum values may be examined. This divide-and-conquer technique may continue until the search reaches atomic values (determined by the gap value of the range). If some atomic values and ranges seems to fulfil the constraints they should be returned. And the subdomain now consists of both ranges and atomic values.

Optimal/promising fixed budget computation

The library could take any combination of constraints and then organise them into conjunctions of disjunctions, with the constraints taking fewer values first.

In the implementation this is done by hand, so the program takes conjunctions of disjunctions of constraints organised with constraints that takes fewer values first. Constraint weighing could also help organising the disjunctions and furthermore lead the search to concentrate on variables that is bound by these constraints. (p. 222 AIRussel).

Constraints should point to specific variables, this makes it somewhat rigid/ineffective because I have to write at global constraint that accounts for everything (ineffective in propagation – might also be a problem if it does not show progress in changing values, i.e. it is a hard constraints and not returning a cost of the set of values.) or divide it into smaller constraints separated by an 'or' (v). The latter is ineffective because there would should be a combination of constraints accounting for every situation, e.g. if the first variable is satisfying an unary constraint, the next say 3 variables (if the problem holds 4 variables) could satisfy three unary constraints, an unary and a binary (two combinations exists) or a constraint that takes three variables. This grows fast with the number of constraints.

Does it make sense that a continuous range cannot have specific values removed? Should it be possible for it be divided into subranges if the user decides to remove a range of values in between its domain of [min;max]?

Pool of workers to compute sections and send results to another worker to handle front page articles. Or, a lazy load approach where the front page is computed and then sections are computed with hard constraints that manages placement of articles within the given sections, e.g. if an article from the front page has the potential to be placed in only one section the constraint should state this, but it would demand cross-worker-constraints to handle if an article from the front page has the potential to be placed in more section (i.e. `xor`). color

Reducing constraints to binary constraints.

6 EVALUATION OF THE SOLUTION

Evaluate using the in [Diaz *et al.*, 2001] and [Esteban *et al.*, 2000] presented method. Their categorisation based on few keywords (5 as the lowest) to represent a category resulted in poor evaluation, this gives a good motivation for including more keywords and using Wordnet to enrich the set of keywords.

It could be interesting to evaluate the precision and recall points based on: news items per section, news items per category, maximum number of news items per message required by the user, general relevance of the contents of a given day for a given user, etc. as [**?**, **?**] proposes.

[**?**, **?**] also raise the problem of precision in finding news within based on a single day. This can hopefully be handled by having the user specify in which period of time he wants news and maybe notify the user that solutions might be inaccurate if a limited period of time is chosen, or just limit the user to specify 24 hours as a minimal value.

6.0.1 *Initial Test*

NOT ONLY 5 test persons [Nielsen, 2012]. This has been discarded by many.

6.0.2 *Result*

6.1 TEST

6.1.1 *Layout*

columns constraints, white space and odd placement of articles.

In order to come closer to understanding what newspapers does it could be interesting to analyse their component structure, e.g. using [Liu *et al.*, 2001] algorithm.

6.1.2 *Content*

The system can be used for automatic classification of articles. Of cause, then a sufficient list of categories and their definitions must be used. This can either be retrieved by the list of Google News categories[1] and a Wordnet enriched list of key words from Google News list of suggested keywords[2] or by the root terms presented in [Abuzir and Vandamme, 2002]. These can later on be refined by information retrieved by the user behaviour in the system and manually removal of false negatives? However, also more advanced techniques of text classification could be used in later stages of the system, like one presented in [Esteban *et al.*, 2000].

Maybe find a better example of text classification.

Use of automatic generation of personal item summaries [Díaz and Gervs, 2005]

Use geotargeting to supply local news.

Use a thesaurus and predefined root terms as in [Abuzir and Vandamme, 2002] which improves classification; semantic knowledge is more general than keywords.

6.1.3 *Functionality*

Order a print copy of the newspaper

The development of the Internet from a distributor of information to a library of digital applications has deeply integrated the users in every step of an applications lifetime. It has become harder to distinguish between super users and developers, applications are branched and modified according to every need and authors can therefore no longer predict which use his application can be to another user – nor should he have to.

1 http://support.google.com/webmasters/bin/answer.py?hl=en&answer=42993.
2 http://support.google.com/news/publisher/bin/answer.py?hl=en&answer=116037.

6.1.4 *Improvements*

[Centeno *et al.*, 1999] suggest virtual communities, or individuals with common interests.

Weights on key words should be adjusted by a strength (or an uncertainty) of prediction as it is proposed in [Claypool *et al.*, 1999].

Count the number of sources have included an articles that are very similar to find the breaking factor.

How to handle that a user is not presented with an already read article? [Billsus and Pazzani, 2000] presents the nearest neighbour (NN) algorithm approach, using a tf-idf similarity, to determine whether the story is already known, i.e. the similarity to the NN read story is above a given threshold. This could be solved by keeping a library of read items (this can be done along with the tracking of which article is in focus) and then match new items against this banned list and down prioritise them if their similarity is too high. This could be done with the same polynomial function as used between articles.

"Distinguishing between short-term and long-term models has several desirable qualities in domains with temporal characteristics (Chiu and Webb, 1998)." [Billsus and Pazzani, 2000].

7 CONCLUSION

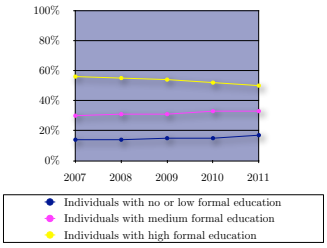
Part III

APPENDIX

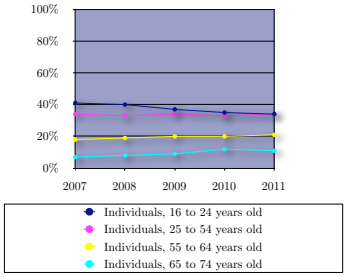
A USER NEEDS

This section will define the user needs for the application.

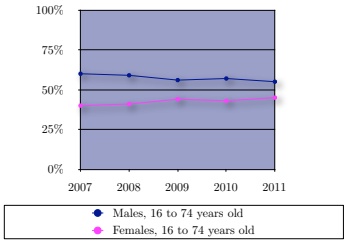
A.1 PERSONAS



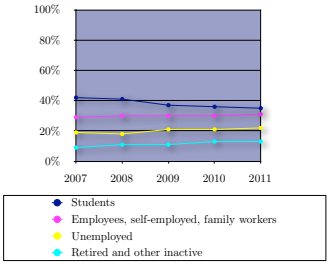
(a) Initial prototype layout with adjustable ratios between articles and a paged interface of each section.



(b) Second iteration of the prototype with an "endless" layout. Sections are placed beneath each other.



(c) Third iteration of the prototype with a column-based and "endless" layout. Sections are placed beneath each other.



(d) Third iteration of the prototype with a column-based and "endless" layout. Sections are placed beneath each other.

Figure 8: Eurostat: Individuals using the Internet for reading / downloading online newspapers / news magazines

In Figure 8 is shown the basis for the division into the following user groups:

- `student` 35%, `employeesself-employedfamilyworkers` 31%, unemployed, retired or other inactive

- `highformaleducation` 50%, medium formal education, no or low formal education
- `male` 55%, female
- `16-24` 34%, `25-54` 34%, 55-64, 65-74

These provide the basis for the following personas.

A.1.1 *Thomas: student medium formal education male of age 21*

Thomas is 21 and a student at the Technical University of Denmark to be a bachelor of engineering in software. He is very interested in soccer and is therefore always updated on sports news. He reads about it online, newspapers and talks about it with friends. With big events he even likes to post it on Facebook. As a soon-to-be software engineer he has a natural thirst for news about technology, and he mainly reads these at home at the dormitory. wired.com, newz.dk, engadget.com, facebook.com computer, Samsung Galaxy Tab

A.1.2 *Laura: employed high formal education female of age 39*

Laura is 39 and is employed as a key account manager. She likes to be updated on strategies and economical status of rivalling companies. She is also very interested in politics and likes to discuss this subject with her friends. She reads economical news and likes to be updated on the run. b.dk, borsen.dk, twitter.com iPhone, iPad

A.1.3 *Marie: unemployed no or low formal education female of age 61*

Marie is 61 and a currently unemployed housekeeper. She spends her day looking for a job and taking care of her pet cat until her husband comes home. She mostly looks for the gossip sections or news about crime or big disasters. She also spends some time reading through the travelling guides as she dreams of going away with her husband. ekstrabladet.dk, bt.dk, nyhederne.tv2.dk computer, Lenovo IdeaPad A1

A.1.4 *Carl: retired or other inactive high formal education male of age 69*

Carl is a retired professor in psychology. He likes to discuss human behaviour and relation with his acquaintances and is very interested in cultural events. Therefore he often seeks the cultural sections and discussion fora to see what is going on. politiken.dk, aok.dk, dr.dk computer, iPad

A.2 SCENARIOS

A.2.1 *Thomas*

Thomas comes home after a day at the study, picks up his tablet computer and opens Editor from the desktop. Editor opens and shows him the front page where all the headlines stories are displayed. The main story is about a new version of the Android OS that has been released today and presses it to read more. The story opens in a full window display with quality images to match the articles. He reads the first section and feels satisfied with the amount of information, but wants to share the information on Facebook, so he clicks share button and writes a comment and posts it on his Facebook wall. He closes the article and returns to the front page. He sees a top story below the main story about Mr. Mærsk Mc-Kinney Møller who has died. It is not a story that falls into his key interests, but as the news is big he is satisfied that he got informed about it. Thomas feels like reading more about technology so he opens the menu and chooses the "Tech" section he has installed in the application. The section opens with a head line and a page number to let him know where in his paper he has navigated to and finds an article about a new multicore CPU technology. He has never been interested in CPU technology before, but finds this technology interesting after reading about it, so he opens the application settings and types in keywords about the technology under his "Tech" section to keep him updated about it. He also adjusts the ratio between general and personal news, to be less personal as he feels like he needs to broaden his horizon a bit with respect to news. He closes the settings menu and Editor immediately starts updating the articles. Some new articles about CPU technology has been included amongst the articles in the "Tech" section after paging through the section and reading some of the most interesting articles he closes the application.

It could be nice if the key words of a story could be or is already highlighted, so he can click it and add it to his positive or negative list.

A.2.2 *Laura*

Laura is on the train on her way to a business meeting this morning and pulls out her tablet and sees she has one notification from Editor. She opens Editor to get updated on todays news. The front page is displayed and there are headlines from different top articles and a notification is shown in the corner. She presses the notification and the pages turns to show her the article, which opens in full screen. After reading it she wants to see todays headlines, so she presses the back button to return to the paper and presses the return to front page button and the paper turns pages to reach the front page. She scans the page to see if there is any big news about her rivalling companies. There is no breaking news, so she just turns the page to browse the content of todays paper. As she browses the "Politics" section of her paper she finds an article about the Prime Minister introducing a new bill about a toll ring around the capitol city. She chooses the article and it is shown in full screen. As she reaches the bottom of the article she sees the comments about it where her friends and most others are against it. She decides to join the discussion and posts a comment on the article wall. She also sees one of her friends has not commented on the article wall and decides to share the article with her as she thinks she would agree with her opinion. She presses the share button and chooses the Editor logo. A list of her friends is shown, some of them who has already read the article is greyed out, but the one she was looking for is not. So she chooses her and a notification is sent to her.

A.2.3 *Marie*

It is morning and Marie wants to check the news with her coffee in the couch, so she opens Editor from her tablet to get updated. The front page is displayed with a collection of stories as highlights of the content of the paper. It mainly contains stories about celebrities and a big disaster that has happened in japan, but there is also a story about a big political change, that she does not find interesting. So she goes to the settings menu and types in "politics" to add to her negative list. She also adjusts the personal/general news ratio to contain only personal news as she wants only news that is directed to her. She returns to the front page which is now free of political stories.

Her newspaper contains many images and videos as she has set her graphical/textual content ratio more towards graphical content.

A.2.4 *Carl*

Sunday morning Carl wakes up, puts over the kettle to make a cup of coffee. While he waits for the water to boil he picks up his iPad and opens Editor to check the news. The front page opens with headlines from the different sections. There is a review article about a new show in the theatre. Carl presses the article and the system turns pages to the "Cultural" section of his newspaper and opens the article in full screen. Because the show gets good ratings he decides to order some tickets to him and his wife, which he does using the devices browser. After this he reopens Editor which opens in a display of the same article, as he left it. Carl pours his coffee and turn to the "Funnies" section. The section displays snippets of comics strips and some humorous and opinionated articles.

A.3 BUSINESS CASE

A.3.1 *Need*

User value: personal quality and up-to-date stories enriched with quality images. This means that content providers should be chosen/verified. Same navigation as actual newspapers, but faster and with endless more content. Instantly up-to-date. Adaptive layout. Adjustable user profile.

A.3.2 *Approach*

personalised content + composition.

Constraint Programming: fast computation - good for optimal solutions, describes the generic solution in stead of how to solve or find it, very easy to tailor the problem definition of the solution and adjust it and even let users make the adjustments - transparency.

Content providers can get to know their readers preferences better and improve the provided content.

A.3.3 *Benefit Per Cost*

Revenue flow: Content providers are paid. Income from advertisers (scattered [Ovesson and Wikström, 2005, p. 6-7]) and users. Income from selling user behaviour patterns and targeted commercials.

A.3.4 *Competition*

FlipBoard, Wired magazine, Zite and app with actual editors affiliated.

A.4 REQUIREMENTS

The above scenarios, user needs and business case led to the following requirements.

A.4.1 *Non-functional Requirements*

- "the clear overview of content, including a beginning and an end, the ease of use, typography and design" [Ihlström *et al.*, 2004, p. 7]
- both general and personal news (collaborate filtering solves that some news are not received, but are universally interesting [Díaz and Gervs, 2005])
- familiarity in design from printed paper [Ihlström *et al.*, 2004, p. 7]
- Design and layout from printed newspaper [Åkesson *et al.*, 2005]
- both images and videos - test
- a good ratio of graphical and textual - test
- front page should give a good overview of the content - test

- "news valuation, e.g. positioning of lead story" [Ihlström *et al.*, 2004, p. 7]
- mobility [Ihlström *et al.*, 2004, p. 7]
- continuous updates [Ihlström *et al.*, 2004, p. 7]
- "easy and intuitive navigation" [Ihlström *et al.*, 2004, p. 7]
- add video and sound [Ihlström *et al.*, 2004, p. 7]
- incorporate social community and social networks

A.4.2 *Functional Requirements*

- "open, turn pages, chose article, read and return" [Ihlström *et al.*, 2004, p. 6]
- section headlines [Ovesson and Wikström, 2005, p. 6-7]
- article headlines
- article summaries / extracts [Díaz and Gervs, 2005]
- menu w. section headlines [Ovesson and Wikström, 2005, p. 8]
- page numbers [Ovesson and Wikström, 2005, p. 6-7]
- press "like" or key word based user profile (mark self or highlighted? right click to add): positive + negative list (keywords+categories [Abuzir and Vandamme, 2002], [Díaz and Gervs, 2005] and [de Buenaga Rodríguez *et al.*, 2004])
- full screen display of article
- organise into personalised sections
- opens in front page view (summery of newspaper 8 articles) [Ovesson and Wikström, 2005, p. 8]
- adjust variables
- share directly (grey out the ones who have read it)

- comment
- see friends comments
- "The presentation schema – headline, abstract, and text, together with a relevance value with respect to the user profile – rates the highest in terms of user satisfaction, and yet it is not the most frequent." [Diaz *et al.*, 2001]
- ability to search [Ihlström *et al.*, 2004, p. 7]
- Landscape + portrait [Ovesson and Wikström, 2005, p. 6-7]
- touch screen interaction [Ovesson and Wikström, 2005, p. 6-7]
- Functionality from online newspaper [Åkesson *et al.*, 2005]
- Name of columnist [de Buenaga Rodríguez *et al.*, 2004, p. 4]
- Transparency of implicit relevance feedback (see/modify current weights of categories) [de Buenaga Rodríguez *et al.*, 2004, p. 7]
- dynamic short-term + static long-term user profile [Abuzir and Vandamme, 2002], [Díaz and Gervs, 2005] and [de Buenaga Rodríguez *et al.*, 2004]
- relevance feedback [Abuzir and Vandamme, 2002], [Díaz and Gervs, 2005] and [de Buenaga Rodríguez *et al.*, 2004]

BIBLIOGRAPHY

[Abidi and Chong, 2004] Syed Sibte Raza Abidi and Yong Han Chong. Constraint satisfaction methods for information personalization. Technical report, Dalhousie University, Canada and Universiti Sains Malaysia, Malaysia, 2004.

[Abuzir and Vandamme, 2002] Yousef Abuzir and Fernand Vandamme. E-newspaper classification and distribution based on user profiles and thesaurus. Technical report, University of Ghent, 2002.

[Apt, 2006] Krzysztof R. Apt. *Principles of Constraint Programming*. Cambrige University Press, 2006.

[Billsus and Pazzani, 2000] Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. pages 147–180, 2000.

[Bird *et al.*, 2009] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 2009.

[Blei, 2011] David M. Blei. Introduction to probabilistic topic models. Technical report, Princeton University, 2011.

[Bouras and Tsogkas, 2010] Christos Bouras and Vassilis Tsogkas. W-kmeans: Clustering news articles using wordnet. pages 379–388. Springer Berlin / Heidelberg, 2010.

[Bush, 1945] Vannevar Bush. As we may think. *The Atlantic Monthly*, 1945.

[Centeno *et al.*, 1999] Vicente Luque Centeno, Mª Carmen Fernández Panadero, Carlos Delgado Kloos, Andrés Marín López, and Carlos García Rubio. Personalizing your electronic newspaper. Technical report, Universidad Carlos III de Madrid, 1999.

[Claypool *et al.*, 1999] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborate filters in an online newspaper. Technical report, Worchester Polytechnic Institute, Massachusetts, 1999.

[de Buenaga Rodríguez *et al.*, 2004] Manuel de Buenaga Rodríguez, Manuel J. Maña López, Alberto Díaz Esteban, and Pablo Gervás

Gómez-Navarro. A user model based on content analysis for the intelligent personalization of a news service. Technical report, Halmstad University, 2004.

[Diaz *et al.*, 2001] Alberto Diaz, Pablo Gervas, Antonio Garcia, and Inmaculada Chacon. Sections, categories and keywords as interest specification tools for personalised news services. *Online Information Review, OIR*, 2001.

[Dub, 2012] Dublin Core Metadata Initiative. *Dublin Core Metadata Element Set, Version 1.1*, 2012.

[Díaz and Gervs, 2005] Alberto Díaz and Pablo Gervs. Personalisation in news delivery systems: Item summarization and multi-tier item selection using relevance feedback. Technical report, Universidad Complutense Madrid, 2005.

[Eirinaki and Vazirgiannis, 2003] Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. Technical report, Athens University of Economics and Business, 2003.

[Esteban *et al.*, 2000] Alberto Esteban, Pablo Gómez-Navarro, and Antonio Jiménez. Evaluating a user-model based personalisation architecture for digital news services. In *Research and Advanced Technology for Digital Libraries*, pages 259–268. 2000.

[Esteban *et al.*, 2001] Alberto Díaz Esteban, Manuel J. Maña López, Manuel de Buenaga Rodríguez, Jose María Gómez Hidalgo, and Pablo Gervás Gómez-Navarro. Using linear classifiers in the integration of user modeling and text content analysis in the personalization of a web-based spanish news service. Technical report, Universidad Europea de Madrid, Universidad de Vigo and Universidad Complutense de Madrid, Spain, 2001.

[Haskins, 1965] Jack B. Haskins. The editorial mix: One solution to a magazine editor's dilemma. Technical report, Journalism & Mass Communication Quarterly, 1965.

[Ihlström *et al.*, 2004] Carina Ihlström, Maria Åkesson, and Stig Nordqvist. From print to web to e-paper - the challenge of designing the e-newspaper. Technical report, Halmstad University, 2004.

[Liu *et al.*, 2001] Fei Liu, Yupin Luo, and Masataka Yoshikawaf Dongcheng Hu. A new component based algorithm for newspaper layout analysis. Technical report, Tsinghua University, Beijing, 2001.

[Lloyd, 1994] JW Lloyd. Practical advantages of declarative programming. Technical report, University of Bristol, 1994.

[Mobasher, 2007] Bamshad Mobasher. Data mining for web personalization. In *The Adaptive Web*, Lecture Notes in Computer Science, pages 90–135. Springer Berlin / Heidelberg, 2007.

[Nielsen, 1997] Jakob Nielsen. Reading on the web. http://www.useit.com/alertbox/9710a.html, 1997.

[Nielsen, 2011] Jakob Nielsen. Top 10 mistakes in web design. http://www.useit.com/alertbox/9605.html, 2011.

[Nielsen, 2012] Jakob Nielsen. How many test users in a usability study? http://www.useit.com/alertbox/number-of-test-users.html, 2012.

[Nilsson, 1999] Jørgen Fischer Nilsson. A conceptual space logic. Technical report, Department of Information Technology Technical University of Denmark, 1999.

[Ovesson and Wikström, 2005] Fredrik Ovesson and Kristin Wikström. From visions to specification - using user designed mock-ups for envisioning user requirements for the future e-newspaper. Technical report, Halmstad University, 2005.

[Perkowitz and Etzioni, 2000] Mike Perkowitz and Oren Etzioni. Adaptive web sites. *Commun. ACM*, pages 152–158, 2000.

[Russell and Norvig, 2003] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Mordern Approah*. Prentice Hall, 2nd edition, 2003.

[Shapira *et al.*, 2009] Bracha Shapira, Peretz Shoval, Noam Tractinsky, and Joachim Meyer. epaper : A personalized mobile newspaper. Technical report, Ben-Gurion University of the Negev, 2009.

[Stanicek, 2010] Petr Stanicek. Color scheme designer, 2010. v3.51.

[Vossen, 2005] M.P.H. Vossen. Local search for automatic playlist generation. Master's thesis, Technische Universiteit Eindhoven, 2005.

[Åkesson *et al.*, 2005] Maria Åkesson, Carina Ihlström, and Jesper Svensson. How would you like your e-newspaper? – converging the best from two worlds. Technical report, Halmstad University, 2005.