

Utilization and Analysis of Layout Characteristics in Document Layout Automation Techniques

Terje Krogstad



Masteroppgave
Master i Teknologi - Medieteknikk
30 ECTS
Avdeling for informatikk og medieteknikk
Høgskolen i Gjøvik, 2009

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Utilization and Analysis of Layout Characteristics in Document Layout Automation Techniques

Terje Krogstad

2009/07/01

Abstract

The flow of information is continually increasing due to the ubiquitous use of information technology. Information communication and distribution is highly automated to meet the demand for an unobtrusive information flow. Servers, systems and applications overtake the tasks of gathering, storing, distributing and selecting the diverse information requests, though the task of presenting the information in a communicatively effective manner is still a manual tasks handled by professionals. In certain contexts, such as the distribution of dynamically generated content by web servers, the manual tasks restrict the system from performing as efficiently as possible. At the moment there is a great potential for production efficacy gain by automating the workflow to some extent. The system utilization of a document layout generator may cover some designer functions and thereby reducing task redundancy related to the presentation of information.

Document layout automation algorithms are typically implemented with weighting metrics for the purpose of quantifying layout characteristics, also called layout attributes, of generated layout in the perspective of visual quality. Several different weighting metrics and algorithms have been studied and implemented in the past years, but still they fail to produce layouts matching the general performance of a designer in practice. Literature suggests that many of the weighting metrics, despite their advanced mathematical implementation, lack the implementation of a persistent set of preference attributes. Weighting metrics based on an extended set of preference attributes related to designer preferences may further improve the performance of a document layout automation algorithm. The scope of this thesis is to establish knowledge about the utilization of document layout attribute operators in document layout automation. How do document layout automation system with an extended attribute operator set perform related to layout quality? The study presented in this thesis reveals that implementing operators for the layout characteristics alignment and equilibrium in layout generators doesn't automatically enhance the visually perceived quality of the layout generated.

A second scope of this thesis is to generalize the modeling paradigms derived from existing literature related to document layout automation systems. Several systems have been proposed for solving layout automation related problems, but still there is a lack of literature related to the generalization of layout automation system models. Is it possible to recognize modeling patterns can be generalized and utilized in a wide range of layout automation systems, applications and / or components? We propose a layout automation system modeling framework based on a grounded theory study of related literature. The presented proof of concept system LaG is modeled based on this modeling framework and provides an affirmation of the modeling framework flexibility regarding layout automation technology and architectural modeling patterns.

Acknowledgement

This master thesis report is related to the media technology master programme of Gjøvik University College and is a part of the Master of Science degree of the author Terje Krogstad.

The thesis report is written solely by me, Terje Krogstad, but there has been other contributors guiding me through the thesis. I would like to thank my supervisor professor Rune Hjelsvold for the many useful discussions during the master programme and Frode Volden for helping me analyze the statistical data retrieved from the study. I will also like to thank my better half, Ingvild Jevne, for enduring my time-consuming combination of professional employment and studies.

Contents

Abstract	iii
Acknowledgement	v
Contents	vii
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Topic covered	2
1.1.1 Overview of Layout Automation	2
1.1.2 Operationalizing Visual Parameters in Layout for Automation Purposes	3
1.1.3 Web Document Layout Automation	4
1.2 Keywords	5
1.3 Definitions and terms	5
1.4 Research Problem	6
1.4.1 Research Problem Area	6
1.4.2 Proposed Research Area Problems Suitable for Further Exploration	7
1.4.3 The Research Problem	8
1.4.4 Motivation and Benefits	8
1.5 Research Question and Hypothesis	9
1.5.1 Research Question 1	9
1.5.2 Hypothesis 1	10
1.5.3 Research Question 2	10
1.5.4 Hypothesis 2	10
1.6 Methodological Approach	11
1.6.1 General	11
1.6.2 Study implementation	11
1.6.3 Data Collection	11
1.6.4 Survey Objective	12
2 Review of literature	13
2.1 Document Layout Automation	14
2.1.1 Related Research Structure	14
2.2 Web Document Layout Automation	21
2.2.1 Web Documents as User Interfaces	23
2.2.2 Adaptive Web Documents	24
2.2.3 Constraint satisfaction in web documents	24
2.3 Layout Generator Algorithms	25
2.3.1 Simulated Annealing	25

2.3.2	Genetic Algorithms	26
2.3.3	Greedy algorithms	26
2.3.4	Heuristics	27
2.4	Preference Attributes in Layout Generators	28
2.4.1	General Preference Attributes	28
2.4.2	Balance	30
2.4.3	Equilibrium	31
2.4.4	Cohesion	31
2.4.5	Proportion	31
2.4.6	Density	31
2.4.7	Alignment	31
2.4.8	Regularity	31
2.4.9	Uniform separation	32
2.4.10	Uniformity	32
2.4.11	Symmetry	32
2.4.12	Economy	32
2.4.13	Homogeneity	32
2.4.14	White-space Optimization	33
2.5	Operationalizing Layout Quality	33
2.5.1	Operationalizing Attributes	33
2.5.2	Operationalizing Overall Layout Quality	37
3	Methodology	39
3.1	Synopsis	40
3.1.1	Study Objective	40
3.2	Participants and Sampling	41
3.2.1	Participants	41
3.2.2	Sampling	41
3.2.3	Rationale	42
3.3	Survey implementation	43
3.3.1	Practitioner Interview	43
3.3.2	Layout Survey	44
3.4	Technological Implementation	47
3.4.1	Proof of Concept	47
3.4.2	Document Layout Generator	49
3.4.3	Technical Implementation of Layout Generator in the Survey	64
3.5	Data Collection	64
3.5.1	Data collected	65
3.5.2	Collecting Data	66
3.5.3	Persisting Data	67
3.5.4	Handling Data for Analysis	67
4	Layout automation modeling framework	69
4.1	Layout automation system modeling	70

4.2	Modeling framework	70
4.3	General document layout system model framework	71
4.3.1	Layout generator	72
4.3.2	Layout optimizer	73
4.3.3	Layout evaluation manager	73
4.3.4	Layout persistence manager	74
4.3.5	Constraint solver	74
4.3.6	Layout model to document mapping	74
4.3.7	Layout automation system API	74
4.4	Data	75
4.4.1	Information structure	75
4.4.2	Layout structure	76
4.4.3	Output document layout	76
4.4.4	Evaluation weighting scheme	76
4.5	Extended modeling	76
4.5.1	Context analysis	77
4.5.2	Expert evaluation	77
5	Results	79
5.1	Practitioner Interview	80
5.1.1	Efficiating work tasks in the process of composing layout	80
5.1.2	The use of well-established tools in the process of composing layout	81
5.2	Practitioner Evaluation of Generated Layouts	81
5.2.1	System Evaluation vs Practitioner Evaluation	81
5.2.2	Practitioner Evaluation	83
5.2.3	System Evaluation	83
5.3	Proof of Concept System	85
5.4	Results Related to Hypothesis 1	85
5.5	Results Related to Hypothesis 2	86
6	Analysis	89
6.1	General	90
6.2	Review of Results	90
6.2.1	Research Question 1	91
6.2.2	Research Question 2	92
6.2.3	System Evaluation	93
6.2.4	System Evaluation vs Participant Evaluation	94
6.3	Summary and Conclusions	94
6.3.1	Hypothesis 1	95
6.3.2	Hypothesis 2	95
6.3.3	Conclusions	95
	Bibliography	99
A	Technological implementation	103
A.1	Code	103

B Document Layout samples	105
B.1 Randomly generated samples	105
B.2 WeightMap samples	106
C Survey User Interfaces	107
D Survey implementation	109
D.1 Survey results, database dump	112

List of Figures

1	960 grid system for web design	3
2	TED: Context based dynamic web page layout	4
3	Lok and Feiner, balance-based layout generation	7
4	Gonzalez et al, web newspaper layout optimization	8
5	Grid-based layout automation by Feiner et al	14
6	Jacobs et al, grid based layout automation samples	15
7	Johari, automatic pagination in catalogue layout	18
8	Kamba et al, the Krakatoa chronicle system model	19
9	Ahmad, IDEAL research framework user interface	20
10	Gajos and Weld, SUPPLE system generated user interfaces. From left shows default, preference-based and ability-based generated user interfaces.	21
11	Kroener, the DesignComposer system architecture	21
12	Kamba et al, the Krakatoa Chronicle	22
13	Zhang and Ren, CATER web interface layout generator sample	23
14	Bateman et al, constraints-based layout	25
15	Lok and Feiner; a sample layout generated by layout automation component BalanceManager	30
16	Bauerly and Liu pixel-based attribute analysis (Symmetry). Perfect symmetry score except for comparing the pixels C3 and D3	36
17	LaG research framework layout automation system	48
18	LaG, system research framework object model	49
19	LaG, system research framework object model	51
20	Zhang and Ren, automated layout framework for Web User Interfaces	53
21	Lok and Feiner WeightMap downsampling for analysis based on pyramid downsampling technique used in image processing	57
22	The research framework persistence ER. (The additional attributes of entity evaluation are described in section 3.5.1	62
23	Sun, general java MVC-pattern model [1]	63
24	Document layout automation system model framework	71
25	Fu and Liu, a gateway from HTML to XML	76
26	Ahmad, IDEAL system model	77
27	Participant overall evaluation of the two modes tested	90
28	Participant attribute evaluation of the two modes tested	91
29	Samples generated by the prototype layout automation system LaG	105
30	WeightMaps generated by the prototype layout automation system LaG	106
31	Initial survey user interface	107

32	Survey evaluation user interfaces. Layout samples generated from basic system mode.	107
33	Survey evaluation user interfaces. Layout samples generated from extended system mode.	107

List of Tables

1	Examples of spatial and abstract constraints semantical tokens	16
2	Layout automation system interface technologies	72
3	System overall evaluation vs participant overall evaluation	82
4	System evaluation vs participant evaluation	82
5	Participant evaluation vs participant evaluation	84
6	Background variables vs system and participant evaluations	85
7	Evaluation of balance of samples generated in the two system modes	85
8	Total score (Sum of all participants' attribute evaluations) of basic implementation vs extended implementation for every layout	87

1 Introduction

1.1 Topic covered

1.1.1 Overview of Layout Automation

This thesis mainly focuses on the research area of document layout automation and document layout automation systems. In general the term document layout automation refers to the principle of automating the process of generating a layout of a graphical information presentation. This process is mainly handled by professionals in disciplines as graphical design, typography or visual communication. Document layout is a part of the theoretical area graphical design, which refers to several professional disciplines such as typography, media design, web design etc., all of which involves practicing design in the perspective of visual communication. The practice of creative disciplines related to design are highly characterized by work tasks of intuitive and creative nature. Thereby the automation of such tasks is dependent on conforming to and modeling the processes of human cognition. Document layout automation is a research field related to scientific areas as HCI (Human-computer Interaction), AI (Artificial Intelligence) with practical application related to the production work flow of general graphical design.

Design practitioners already benefit from the use of tools that supports and automates the task of creating a graphical design. Graphical designers interviewed in this project emphasizes the professional advance related to utilization of support tools such as templates and grid systems to assist the task of laying out graphical elements. The designers amaze themselves when reflecting over how many automation tools they really are reliant on when practicing design. Automation tools for specific tasks are already highly incorporated in typical software used in the design industry (Adobe software, Quark software etc.). These software packages provide functionality for automation of tasks as aligning elements, regular spacing between elements, snap to guides for consistent positioning, automatic distribution of text content in columns and paragraphs etc. Other more advanced and often vendor-specific automation tools are “step and repeat” for repeating series of graphical elements, slicing web designs for html export, recording events and actions for repetitive tasks, using programmed macros etc.

The continuous development and implementation of automation functionality in designer software is clearly affecting the design production workflow. Still the industry relies on manual labor for the creative tasks. The information technology era that we are a part of has greatly influenced the general flow of and access to information. [2] The effective information flow and the ease of access to the information are two major causes for the increasing amount of information available, [3] [4] but the efficiency of the information flow can be further enhanced; the human as a part of the information flow merely plays an obtrusive part of the potential efficient interchange of digital information between systems and thereby denying information systems to perform up to their capacity. The designer is still an irreplaceable part of the information presentation work flow; the complexity of the design task is at the moment far beyond the capacity of the machine. Although the machine performance is adequate enough to simulate some of the tasks performed by the designer and the task of laying out graphical elements may be one of these.

The thesis will provide a deep dive into some of the core building blocks of layout automation systems. Several different approaches have been presented regarding the technological implementation of automation tools and systems, and the research area is constantly evolving. We

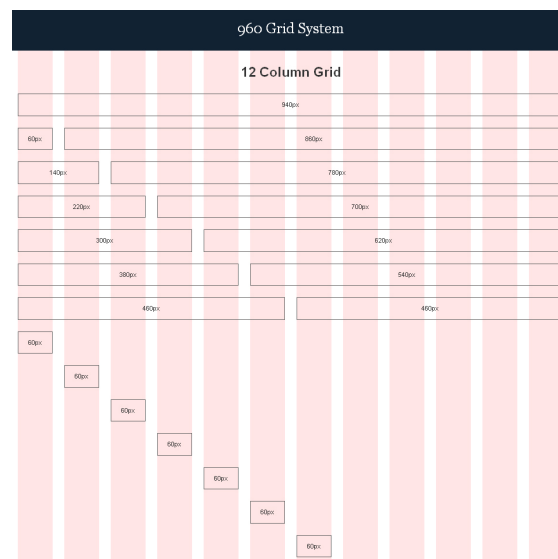


Figure 1: 960 grid system for web design

will be focusing on how utilization of visual preference attribute operators inflict on the general quality of a document layout. This involves operationalizing typical visual characteristics of document layout such as balance, equilibrium and alignment; how can these operators be utilized in an automation perspective and how do these operators influence the quality of the generated layout within the context of a fully implemented layout automation system.

1.1.2 Operationalizing Visual Parameters in Layout for Automation Purposes

The characteristics / attributes of a page layout can be considered as visual parameters. Balance, alignment, symmetry etc. are examples of visual parameters. We denote the layout visual characteristics as layout attributes. Operationalizing and utilizing layout attributes in layout generators is practically implemented as a function of the quantified evaluation of these attributes. There has been proposed different solutions for both quantifying identified attributes of page layout and metrics for operationalizing these attributes suitable of utilization in a technological implementation context.

According to design theory many typical attributes of layout may be derived for the purpose of operationalizing them as parameters suitable for utilization in layout automation. This aspect of layout has been extensively studied and a substantial collection of attributes have been presented in different research articles, both as mere characteristics of visual information presentations and operationalized as potential parameters in a layout automation system metric. [5] [6] [7] [8] [9] A layout generator is dependent on working with quantified parameter representations of these attributes to be able to evaluate the quality of a generated layout. Typical attributes of a layout can be balance, regularity, alignment, cohesion, equilibrium, symmetry etc.

The implementation of metrics that evaluate different parameters of layout naturally varies depending on which attribute to operationalize. E.g. there are several different proposals of how

to operationalize the layout specific attributes balance, alignment, white-space etc. Although the metrics may differ, the layout results are always an optimization of the given attribute or an optimization of the function of the constellation of n attributes. The evaluation of attributes in a layout perspective is an algorithmically complex optimization problem, were the final layout result usually is a local optimization due to the large solution search space. A typical challenge of large search spaces is to end up with a local optimization as near as possible to a global optimization within this search space. [10] [4]

There have been several proposals and approaches for defining useful metrics for operationalizing attributes in the perspective of layout automation. Some of the proposed solutions are formed as mathematical descriptions of the specific attribute [11] [8] and some are merely descriptions of the attribute characteristics [5]. Others have focused on operationalizing in a programmatic context, e.g. by proposing a practical implementable solution for e.g. a balance operator. [9] [6]

1.1.3 Web Document Layout Automation

Web document layout automation is a specialization of general document layout automation. Distribution of information through the world wide web is growing in popularity. In a business perspective, small or large, the use of web as both a marketing channel and business transaction channel is more or less essential. The dynamics of the web enables a broader dissemination of information and ease of communication which affect the entire world sociological and cultural. [12] [13] [14] [15] The multimodal use of the web continually push the technological utilization of this media channel, and the convergence of media technologies on the web further enriches the user experience and thereby the usage, which again affects the demand for effective information production through the channel. [13] [16]

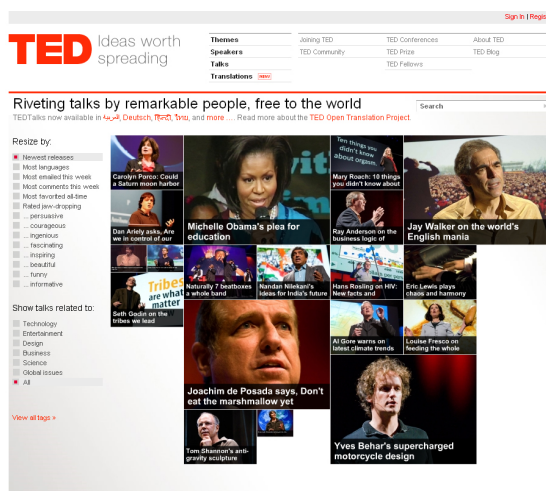


Figure 2: TED: Context based dynamic web page layout

Automation of web document generation is a possible process that could further enhance the web information production efficacy. An implementation of layout automation may be utilized

in different web applications as dynamic product presentation, web newspaper articles presentation, search query result presentation, advertisement presentation etc. All these examples involve elements with related preference attributes (Operationalized in layout automation as abstract constraints) and individual user information preferences on which the elements can be ranked and categorized (E.g. semantically) related to presentation. [4]

There are some differences between general documents and web documents that needs to be considered in layout automation. A typical graphical document normally has a defined surface size on which the layout elements may be layed out on. The surface size is more dynamic in web documents in that it relates to the client viewing medium both horizontal and vertical. [17] [18] [19] First of all the area at hand in the vertical direction is fluid. The vertical height of the information surface will adapt to the content that is displayed and the horizontal width of the display surface will adapt to the client viewing medium (Either the screen size or the window size). On the contrary general documents usually relate to a fixed surface size with a constant width and height (We state “usually” as it is entirely possible to instantly generate general documents with fluid size by using dynamic generation systems, but the client software graphically rendering the document doesn’t dynamically adjust the surface size based on the content and display size)

1.2 Keywords

Layout automation, document layout automation, web document layout automation, layout generator, layout automation system, layout automation system model, modeling framework, evaluation techniques, electronic publishing, simulated annealing.

1.3 Definitions and terms

- *Layout* - The distribution of elements on a defined surface. Graphical layout refers to the distribution of graphical elements on a defined presentation surface, including attributes such as page and type size, typeface, and the arrangement of related elements.
- *Layout automation* - The automation of processes related to the distribution of elements on a surface.
- *Document layout automation* - The automation of processes related to graphical layout.
- *Layout generator* - Software designed to generates layout.
- *Layout evaluator* - Software or component designed to evaluate characteristics of layout.
- *Evaluation manager* - A programmatic manager that administrates the evaluation of a certain aspect of a given problem. In layout automation evaluation managers are implemented to administrate and effectuate the evaluation of attributes of the layout.
- *Layout operators* - Software components operationalizing and quantifying the notion of layout characteristics.
- *Layout attributes* - Characteristics of layout.
- *Preference attribute* - The visual preference of certain layout characteristics.

- *Constraint solving* - The use of mathematical constraints is relevant to many different formalized problems related to computational problem modeling. In layout automation constraints are both related to operationalizing the physical constraints of layout and the abstract constraints of the layout typographical and graphical semiology.
- *Template engine* - Software that is able to process templates with content variables that are replaced with context-dependent dynamic content. Template engines may also support the use of lingo describing logical and functional structures related to programming.
- *Personalized documents* - Documents with content areas both for static content common to several similar documents and variable content. The variable content may be the name and address in a bill, the company logo to the company which a letter is sent to etc. These type of documents are also called “variable documents”. [3]
- *2D-BPP* - The two dimensional bin packing problem; minimization of the number of bins used for packing a set of variable sized objects. [10] [20]

1.4 Research Problem

1.4.1 Research Problem Area

The field of document layout automation has many unexplored areas and unresolved issues. We have picked up the ball passed from diverse earlier research projects related to evaluation techniques; operationalizing layout attributes and utilizing evaluation of a set of attributes in layout automation. Earlier research have either focused on evaluation techniques related to certain attributes of layout. Lok et al focused on operationalizing and utilizing the balance attribute in the perspective of document layout automation,[9] Purvis et al implemented a set of operators for attribute evaluation and focused on the overall evaluation based on this set rather than dismantling the evaluation attribute by attribute etc. [3] One of the unexplored aspects of operationalizing document layout attributes is the investigation of how the certain attributes affect the totality of the evaluation of a generated document layout. How does one attribute inflict on another? How does the same attribute inflict on the overall evaluation of the layout?

A second scope of this research thesis is general implementation, modeling and application of document format-independent layout automation systems. We propose a layout automation system modeling framework for supporting the modeling of these kind of systems. We also present a proof of concept system related to web document layout which was modeled based on the modeling framework. Web document layout automation relates to many of the basic problems described for document layout automation as the principles are the same; laying out graphical elements on a plane surface. [18] [17] [16] [21]

Due to the development of server-side technologies of application servers and server-side script language systems (PHP, perl, Ruby on Rails, Java Servlet/JSP etc.) the information in web documents are often generated dynamically. Template engines have further reduced the complexity of setting up web application user interfaces. Although the information generation of web documents can be completely dynamic, the presentation of the information is not. Templates are more or less static (One might use complex logical structures and iteration statements

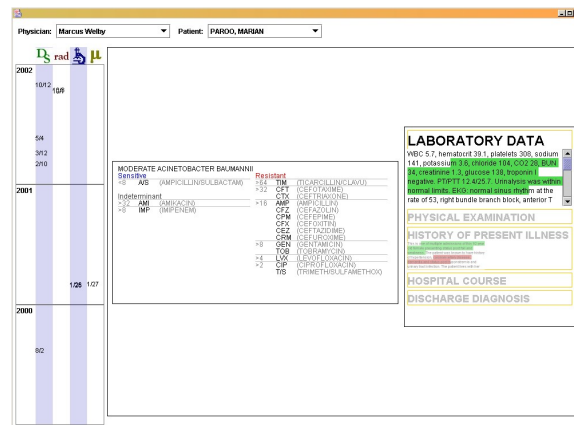


Figure 3: Lok and Feiner, balance-based layout generation

to dynamically generate different layouts) and web document styling technologies as CSS and XSL are more or less static (One might define block elements in style sheets so that they are dynamically rendered related to the display surface size). [22]

Earlier research have focused on the client-side implementation of document layout generation based on client-side scripting technologies as JavaScript; the client browser receives the information as HTML and automatically maps this content to an optimized layout based on script-based DOM-manipulation. [18] [17] [16] In the research we have focused on the aspect document format-independency in layout automation system and based on this we propose a modeling framework for general modeling of layout automation systems.

1.4.2 Proposed Research Area Problems Suitable for Further Exploration

Simon Lok et al suggests a further enhancement of the balance operator to support other types of balance. The balance manager implemented in their research supports crystallographic balance; possible other balance operators would have been radial, symmetric or asymmetric balance. In addition they suggests a further enhancement of the weight map to support per pixel value weight map calculation. [9] Jacobs et al also recognize the need for utilizing metrics to evaluate layout attributes such as balance in template systems for layout automation. [23] Ahmad discuss the demand for incorporating multi-criteria decision making in layout generators and states that implementing metrics for different aspects of layout is no guarantee for improving the total information value of the layout, thus this needs to be explored further.[10] Purvis et al suggests the implementation of trade-off automation related to preference attributes. They discuss the possibility to implement dynamic weighting of preference attributes depending on user / designer priority of different preferences. Within this scope they imply the necessity to further understand how these preference attributes affect each other in the total evaluation of a layout. Bateman et al addresses the insufficient knowledge related to the nature of layout. They describe the lack of a technical description and operationalization of page layout generation. Broad issues related to functionally defining page layout have been left unaddressed as research have tended to focus on local issues of typography and text-formatting. [24] The demand for further exploration

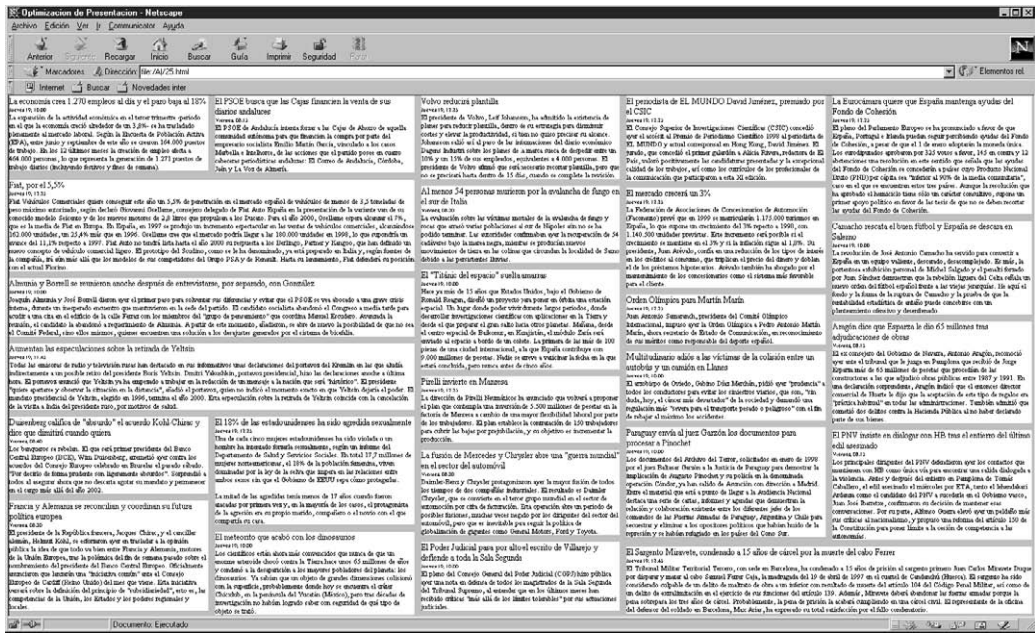


Figure 4: Gonzalez et al, web newspaper layout optimization

of the nature of document layout aesthetic is also problematized by Cai et al as they claim to provide one of the earliest attempts to manipulate dimensions of web document aesthetics in experimental settings. [25]

1.4.3 The Research Problem

Earlier research in the area of document layout automation describes several different approaches to deal with the problem of laying out graphical elements on a display surface. In this research project we have been focusing on some aspects related to evaluation techniques in layout automation. As we described in the last section there is a great need for further exploration of the utilization of attribute operators for evaluation purposes.

1.4.4 Motivation and Benefits

The convergence of traditional technology areas as information technology and media technology introduces interesting areas of new research. The power of information technology tools and the digitization of different media types makes it possible to manipulate traditional media in a totally new manner. Applications of interactivity and automation of media production and consumption emerges as technologies slowly melts together, a development motivating the standardization of seamless media technology solutions.

The general convergence of media technology is ubiquitous and the force of standardization technologies further accelerates this convergence. [2] The production workflow of digital media is highly automated, making the consumer take on the role as the producer, gradually moving into the sphere of the professional. In the light of this information it is natural that the automation of document generation becomes relevant as part of the paradigm. The possibility to

improve efficiency of the work flow of digital documents seems to be an important part of the strategy of graphical businesses.

As technology in the area of information presentation evolves, the information technology industry experience that the ubiquitous access to information through different channels and media devices is a challenge related to how this information is presented. A web document displayed on a large LCD-screen may effectively communicate the information content because the presentation have been designed for this medium. Displaying the same web document on a small mobile device screen may result in communicative problems related to the insufficient design and layout of the information content. [26] Document layout adaption to the physical context of the viewing environment is also a challenge related to document layout automation. Fluid layout and adaptive layout generation are concepts that may provide a solution to these challenges. [19]

Already the digitization of tools and tasks within the design industry has provided a revolutionary way of producing graphical products. The process of layout design is clearly an important part of todays graphical workflow redundancy. This redundancy may be partly eliminated with the use of automation technology. Still the human as a creative designer artist strongly outperforms the machine; but the possibility to automate rule- and convention-based layout is being explored for the purpose of efficiating these processes. The systems for automation of this kind of layout may potentially be improved by enhancing already existing evaluation techniques and establishing more robust metrics for the quantification of quality related to document layout.

Most of the existing prototype systems related to document layout and web document layout show little flexibility related to document format portability. The tools presented in research literature is highly specialized towards specified problems of document layout. A document layout automation system model for document format-independent layout generation would improve the potential benefits of using such systems. The design industry produces design specified for a wide variety of document technologies. The same design product is often distributed in a diversity of channels and a variety of document formats. Existing specialized document format-specific layout generators can't handle format flexibility, thus a general system model document format-independent layout automation system would be of great benefit.

1.5 Research Question and Hypothesis

1.5.1 Research Question 1

Does an extended set of attribute operators in a automatic layout generator strengthen the visual quality of the generated layout?

In earlier research there has been a major focus on solving the problem of operationalizing a narrow aspect of document layout characteristics. E.g. operationalizing only balance, only symmetry etc. Few research projects have focused on how the different attribute operators inflict on each other. Does an extensive set of operationalized attributes automatically mean that the layout generated is qualitatively better? Do the different attributes influence each other in such a way that the layout quality is increased? Maybe the difference isn't noticeable? Or maybe the

layout quality is actually degraded as a result of the preference attributes influencing each other?

The operator balance have been extensively researched and implemented based on several different approaches. The approach for this thesis is to use the experiences made by operationalizing the balance operator in layout automation. [4] [9] [10] We aim to establish knowledge related to the use of such preference attribute evaluations in document layout automation. The thesis is inspired by the work of Lok and Feiner related to the balance operator. An extensive discussion of this work is provided in the chapter 2. Lok and Feiner focuses on weighting balance as the main operator for generating layouts. As an extension to this work we have focused on operationalizing alignment and equilibrium as a supplement to the balance operator. As far as we can see these operators have not been practically implemented in a document layout automation system before. Alignment has been implemented as a constraint parameter, but not as an operationalized attribute with a stand alone evaluator. [3] [9] The purpose of using these operators is to see how they influence the basic system implementation regarding layout quality.

1.5.2 Hypothesis 1

Layout automation system utilization of additional operators for visual parameters strengthens the perceived visual quality of generated document layout. Operationalizing alignment and equilibrium in layout generation increases the perceived visual quality of generated document layout.

1.5.3 Research Question 2

Does a layout automation system model framework for document format-independent layout generation prove to be realistic in a real-world implementation?

Earlier proposed solutions of web document layout generators have focused on a client-side implementation in client-side technologies; mainly javascript. [16] Other implementation propositions are directly related to print document layout automation systems. [9] [3] [24] [23]

Based on the experiences from modeling our layout automation system LaG, an extensive literature review and a thorough examination of architectural modeling and component development presented in the related literature we try to derive paradigms regarding layout automation system modeling. Generally the research literature lack documentation of how layout automation systems are architected, making the theoretical field of layout automation system modeling intangible. This means that every researcher or system engineer entering the field of layout automation needs to model such systems from scratch. By systemizing the architectural and technological challenges related to layout automation systems we aim to provide a layout automation system model framework.

1.5.4 Hypothesis 2

A flexible general model framework of a document format-independent layout automation system is applicable to real-world scenarios.

1.6 Methodological Approach

1.6.1 General

In this study the main purpose is to map how operationalized preference attributes influences the perceived visual quality of a machine generated layout. There are three main aspects related to this kind of study; the operationalizing of quality based on attribute operators, the machine evaluation of these attributes and the overall quality of the layout and the human quality perception evaluation of the same attributes and overall quality. We present a document layout automation system as proof of concept to serve these purposes.

To ensure that the evaluations of attributes related to document layout where consistent we asked professional in the area of design to participate in the study. The participants where chosen from professionals in the design industry. The six chosen experts where to evaluate a total of 25 generated document layouts in the perspective of quality.

Another aspect of this thesis is the layout automation system modeling. We propose a layout automation system modeling framework based on a qualitative grounded theory study of existing research literature and experiences derived from practically implementing a proof of concept layout automation system for use as a research framework.

1.6.2 Study implementation

The study was implemented with two separate sessions; an interview session and a survey session. The purpose of the interview session was to map the current use of layout automation tools and how the participants evaluated the demand for using automation tools. The survey session consisted of the participants evaluating characteristics of presented layout samples produced by the proof of concept system.

The basic implementation of a layout generator depend on utilizing the quantification of layout quality based on a set of operators. As a generator is able to quantify the notion of layout quality, it is also able to evaluate one layout related to another. This means that one can consider the process of evaluation as a function to be minimized or maximized depending on the quantification scale used.

We have implemented a prototype web document layout automation system called LaG. The core layout automation system is possible to run both on a server and as a local application. The layout it produces is a format independent model that can be mapped to any format or technology. It is built to potentially generate layouts for any type of document format. The current solution is implemented with a web API that maps the layout-model to a HTML/CSS-layout. Which format the core layout model is mapped to depends on the application utilizing the system. It is possible to implement an applications that map the layout model produced by the system to different formats and technologies as xml, pdf, latex etc.

1.6.3 Data Collection

Each evaluation generated data related to layout. An evaluation consists of both the participant evaluation and the system evaluation of the attributes of the specific layout. The data was collected through an implemented research framework based on the layout automation system model proposed in this thesis. The participant evaluates the layouts through a web application user

interface. The web application takes care of evaluation data persistence.

1.6.4 Survey Objective

The survey objective is to gather data about layout automation systems based on preference attribute operators and evaluation techniques. In addition the research framework used in the survey is a proof of concept layout automation system implementation based on the proposed document format-independent layout automation system model framework.

2 Review of literature

2.1 Document Layout Automation

Document layout automation refers to the field of layout automation related to information presentations. This is graphical presentations of information elements in viewing media as paper, display screens etc. Page layout (Document layout) is more properly termed typographic design according to Bateman et al. It is divided into three sublevels: Macrotypography, microtypography and style. The principles of macrotypography is closely related to document information block layout and can be considered as the segmentation of information blocks distributed on the presentation surface. [24] Microtypography refers to principles related to legibility, text formatting and text appearance and style refers to overall presentation of information.



Figure 5: Grid-based layout automation by Feiner et al

Lok and Feiner defines layout as *the process of determining sizes and positions of the visual objects that are a part of an information presentation*. The definition isn't entirely precise as layout refers to the generality of laying out elements, either graphical or physical. The definition is adequate to describe the term document layout. Layout automation (Again they are more precisely defining "document layout ...") is defined as *the use of a computer program to automate either all or part of the layout process*. [4]

2.1.1 Related Research Structure

Research related to document layout automation spans from generalizing the principles of layout automation to providing analysis of subproblems as operationalizing document layout characteristics, automatic pagination of generated documents, constraints in layout automation etc.

Document layout automation is a subproblem of the more general (and widely problematized) layout automation problem. The layout automation problem is often formalized as an optimization of the bin-packing problem, a formalization which have been adopted and utilized in research related to areas as document layout automation and VLSI (Very-large-scale integration). [10]

Lok and Feiner provides a useful research survey which nests up the threads of literature rela-



Figure 6: Jacobs et al, grid based layout automation samples

ted to layout automation. They describe and categorize the general techniques used for problems related to document layout automation. [4] The work of Lok and Feiner is of interest related to both the particular research area and related areas as document layout user adaption etc. [10] [27] We have found influence in this work both in the research methodology and the technical implementations of the prototype LaG which serves an important role in retaining knowledge related to the research questions described. [10]

Layout automation techniques

There are many possible technical approaches to the general problem of document layout automation. The technical solutions applied to the problem of aesthetically optimize the layout of a document have been extensively examined by Lok and Feiner. Lok and Feiner divide these techniques in four different categories; simple techniques, constraint satisfaction, learning techniques and evaluation techniques. [4] [9] [27] General methods of implementation related to

layout automation often utilize a combination of the techniques described by Lok and Feiner. An example is the wide use of constraint satisfaction in combination with algorithms related to learning techniques and / or evaluation techniques. [4] [9] [3] [10]

Simple techniques refers to a generality of techniques implemented in existing UI-toolkits and -managers (E.g. window managers administered by operating systems). UI-objects are incorporated with defined constraints and methods to automate the proportion and size. A container UI-object containing several other UI-elements as buttons and text fields etc. define a constrained area on which these elements may be graphically rendered. UI-object methods refers to event handlers on which to act when a user interaction have occurred; e.g. maximize window etc. Another example of simple techniques related to layout automation is the use of templates in WYSIWYG text-editors. Word processors is able to lay out the information persisted in a database based on template labels defining were to put specific content. E.g. a letter template may define were to put the name of the receiver, the address of both receiver and sender etc.

<i>Spatial constraints</i>	<i>Abstract constraints</i>
image1 ABOVE image2	image1 RELATES_TO image2
image2 LEFT_OF text1	text1 DESCRIBES image2
image3 RIGHT_OF text2	text2 FOLLOWS image3

Table 1: Examples of spatial and abstract constraints semantical tokens

It has been a major focus on constraint satisfaction techniques in research related to document layout automation the last decade. Constraint satisfaction refers to the programmatic operationalizing of mathematical principles of linear constraints. The notion of constraint can be divided in two classes; abstract constraints or spatial constraints. Abstract constraint refers to the inter-element level of semantic relation. A product and a price tag have a high level relation regarding semiology and the abstract constraint on this level may imply that these elements should be placed adjacent to each other. [17] [28] [4]

The concept of spatial constraints refers to the mathematical representation of geometrical inter-element relationship. It is fairly intuitive to define that the generator should avoid inter-element overlap mathematically. This particular problem can be expressed as linear inequalities that constraints the left border a_n of one graphical element can't be within the borders a_m and b_m of another element. [17] [28] [4] A practical example of four conditions of constraints expressed for the no-overlap problem as described by Arahim to ensure that block B_i does not overlap block B_j [10]. One of the following conditions must be satisfied:

1. Block B_i is left of block B_j .
2. Block B_i is right of block B_j .
3. Block B_i is below block B_j .
4. Block B_i is above block B_j .

The constraints expressing the four conditions (expressed as linear inequalities), where the parameters denote geometric position, x and y (Bottom left corner of block for ordinary documents and by convention top left for web documents), width w and height h :

1. $-x_i + x_j \geq w_i$
2. $x_i - x_j \geq w_j$
3. $-y_i + y_j \geq h_i$
4. $y_i - y_j \geq h_j$

Learning techniques refers to techniques related to machine learning. An example of learning is the machine adapting layout based on gathered data from user interaction. The system tries to infer the preferences of the user based on his / her use of the interface. [4] [10]

Evaluation techniques refers to the use of quantitative metrics to operationalize the document layout characteristics. The layout generator tries to evaluate the layout quality by measuring certain attributes as balance, alignment etc. Based on the evaluations of the specific attributes it calculates the overall quality based on a proper function. [4]

Operationalizing layout characteristics

Evaluation techniques generally rely on the quantification of layout characteristics. Layout characteristics are commonly related to the layout visual appearance, also frequently denoted as layout attributes, layout preference attributes or visual attributes. Typical characteristics are generally related to spatial attributes of the overall layout, spatial attributes of the graphical elements or the relation between these graphical elements.

The balance of a layout is a typical example of an overall layout attribute; it is operationalized as a measure of the overall distribution of elements on all the available spatial surface. An example of spatial attributes related to a specific graphical element is proportion; an image is well proportioned as long as the it keeps the original aspect ratio. The overall proportion attribute is measured as a function of the individual elements proportion score. Alignment and symmetry are examples of an inter-element attribute; attributes that are derived from specific spatial relations between graphical elements. E.g. the calculation of an overall alignment score is clearly dependent on at least two spatial parameters.

There is no definite objective and quantitative aggregation of the visual quality of a document layout. Terms like visual quality and visual aesthetics are highly subjective prone to be influenced by individual preference. [10] In an evaluation-related technological perspective the measure of layout quality relates to the practical implementation of metrics that quantifies the layout design guidelines from which professionals bases their layout production. [5] [11] [9]

Practical document layout automation implementations

Several practical approaches and system implementations have been presented in the related research literature. Many of the concepts described in this thesis and principles described in other articles on the subjects have related the findings to proof of concept applications and systems either as pure research frameworks or as basis for the deriving new knowledge from practical use cases. This thesis also proposes a document layout automation system modelling

paradigm based on experience from related literature. Thereby it is reasonable to present a short overview of the tangible document layout automation systems described in research literature (Tangible in the sense of accurately named, described and practically utilizable).

Although there have been proposed several systems which solves particular problems related to layout and layout automation (Such as automatic pagination, text wrapping, automatic UI-component functionality, optimization algorithms of related problems etc.), we merely describe the systems influencing our work. The yellow page pagination system of Johari et al is one of the

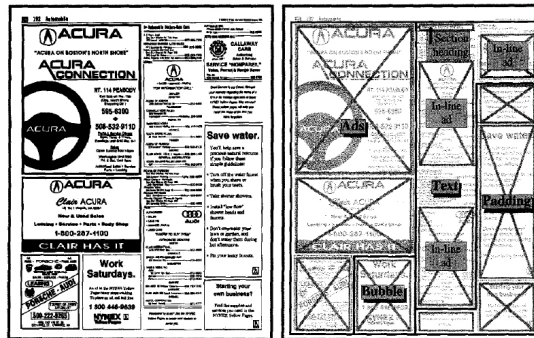


Figure 7: Johari, automatic pagination in catalogue layout

earliest research articles which have directly inspired our work. Johari et al presents a system for automating the pagination and layout of phone book yellow pages (Fig. 7). Generally yellow pages contain a huge amount of information distributed on several pages; improving the efficacy of layout object surface resource usage will generally improve the efficiency of many resource aspects of producing the yellow pages. The article presents methodologies to automatically generate layouts with compact and harmonious positioning of text flow and advertisement elements on the pages. The layout optimization technology used for yellow page pagination is based on heuristic search and a simulated annealing algorithm. [29]

Kamba et al presents an early practical system implementation related to web document layout automation called the Krakatoa chronicle. This is an online newspaper layout automation system based on a client java applet. The client applet uses a server-side application (A perl script) to fetch resources dynamically and updating user profile by tracking the relevance feedback.(Fig. 8) The automation of layout is based on personalization technologies; the system dynamically adapts to the logged user and community preference (A user belongs to a community of users) in addition to general layout preferences. [30]

Bateman et al describe a prototype system used for document layout automation of art history information. The system is a part of a research framework related to testing general methods and technologies for information presentation layout generation in a broader perspective. Bateman et al claim that earlier research (Earlier than 2001) have been restricted to relatively narrow problems related to layout generation. [24] The system provides a practical application of utilizing evaluation techniques related to designer preferences and typical layout characteristics. They also present the principle of layout structure; modeling the general layout for application related to equivalence detection and layout pattern matching.[24] The technological utilization

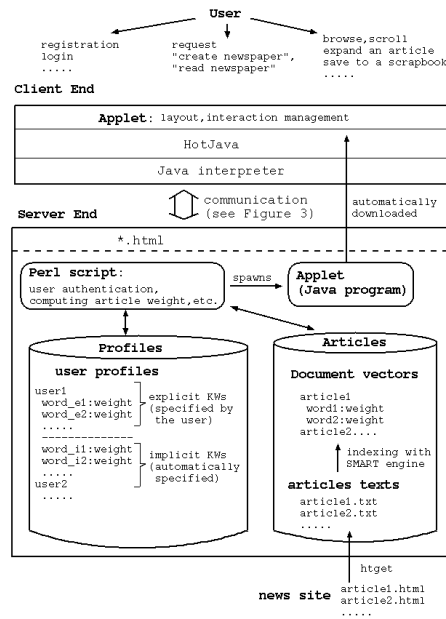


Figure 8: Kamba et al, the Krakatoa chronicle system model

of such principles makes a layout automation system competent to match different layout structures and information structures to detect similarities which is indeed practical in the perspective of generated layout reusability, layout optimization and system heuristics.

Lok et al present the principles of utilizing a balance operator in the prototype layout application BalanceManager. The BalanceManager ensures balancing the layout in both the vertical and horizontal direction inspired by image downsampling techniques (Ref. 2.1). The BalanceManager have been implemented in an existing layout automation system used at the New York Presbyterian hospital for controlling the placement of windows displaying patient record related documents. This real life implementation scenario shows an actual use case of such systems. [9]

Another actual use case and an impressive implementation is the template-based layout automation prototype system presented by Jacobs et al; they've actually used their layout engine to layout the article presenting the research project. The prototype system is based on automatically optimizing layouts based on layout templates authored by the users of the systems. The layout engine evaluates how well the content fits to provided templates and returns a template score. This template score is used by a paginator to optimize the pagination of the layout, which results in an optimal sequence of templates to use for the particular document. [23] The system isn't fully automated as users have to author templates, but the concept addresses principles inspiring the architectural model framework presented in this thesis.

Research related to layout automation is often dependent on research support systems, prototype experimental systems and research frameworks to explore the research field. Ahmad presents a research framework for testing hypotheses related to the 2d-BPP problem and general layout automation problems. The system enables expert layout knowledge representation, for-

mal modeling of user preferences and user controlled layout automation settings such as use of optimization algorithm, number of layout objects etc. [10]

The system was used as a research framework for Ahmad's experiments on layout automation decision support and multi-attribute layout optimization (Fig. 9); the phrase related to the system name acronym IDEAL (Intelligent system for Decision support and Expert Analysis in multi-attribute Layout optimization) precisely describes the system utilization.

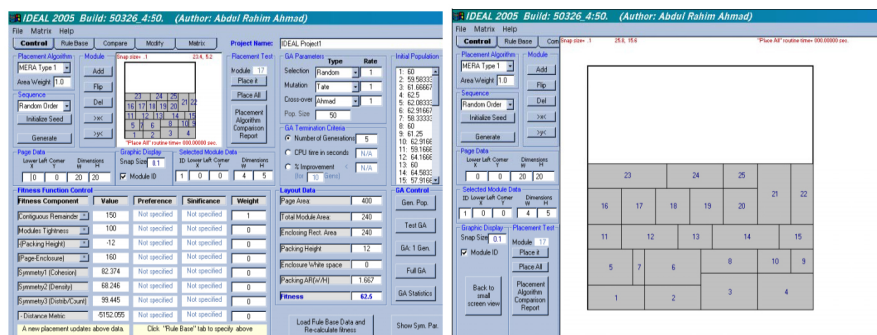


Figure 9: Ahmad, IDEAL research framework user interface

The layout of user interfaces is closely related to general document layout. General design guidelines apply for document layout automation and UI layout automation. Fogarty and Hudson present a UI layout automation system called GADGET for user interface and display generation. The GADGET system is an experimental toolkit designed for experimental purposes and research settings. It enables exploration of optimization principles within a layout context. While GADGET administrates the optimization aspect of the evaluation techniques, the evaluation itself needs to be provided by a separate component. The system is dependent on three separate components; an initializer, iterations and evaluations. The initializer is the initial solution, the iterations is the procedure to run after evaluation and the evaluation is the actual layout phenomenon quantification (E.g. a preference attribute) upon which the system optimizes. The system is generally based on common abstraction and solver paradigms related to several optimization problems, though specializes to UI-related problems by empowering the implementation of user interface property support and a library of reusable components. [31]

The SUPPLE system of Gajos and Weld is a fully usable layout automation system of practical application. The system is designed to automatically generating user interfaces for different display contexts and users of different interaction capabilities.

The novel solution is based on treating interface automation as an optimization problem related to device constraints and user interaction demands. The system has actually been reviewed by the Guardian [32] and an introduction video is available at Krzysztof Gajos' web page [33] at Harvard University (Fig. 10). [34]

The automation of web-based user interfaces has attracted the attention of several researchers. The DesignComposer is a web-based layout automation system developed presented and developed by Kroener. Kroener describes the system as context-based automated layout for the internet.



Figure 10: Gajos and Weld, SUPPLE system generated user interfaces. From left shows default, preference-based and ability-based generated user interfaces.

The system is initiated by a request for a presentation of a certain information structure (XML-based structure scheme). The presentation input structure is forwarded to a layout generator (Fig. 11) which generates a layout based on this information in addition to context-related information requested from a context retrieval module. The layout generator response is a presentation (HTML presentation of the layout) adapted to the client viewing environment. [35]

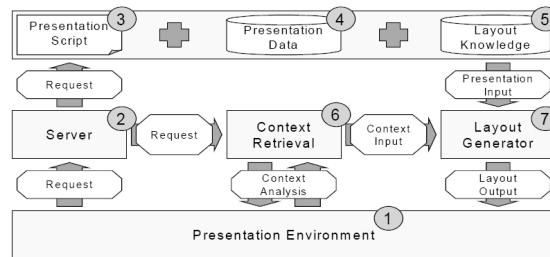


Figure 11: Kroener, the DesignComposer system architecture

Another web-based solution is the CATER-framework presented by Zhang and Ren. The framework is a constraint-based solution for automating the generation of layout of transactional web-pages. They present a novel field-to-widget solution for automatic mapping of database fields to UI-widget layout elements. The system is inspired by the SUPPLE system of Gajos and Weld [34]. Both systems utilize user tracing and user interaction information gathering for user preference layout automation optimization, but the CATER system is specialized for layout automation of web layout with user interaction logging through HTTP. [15]

2.2 Web Document Layout Automation

In the perspective of web document layout automation research have evolved around several specialized cases. The research have attacked problems as optimizing the surface usage in relation to display size, mapping web content to small display sizes, automatic generation of transac-

nal pages etc. Web document layout automation is closely related to general document layout automation. The main difference is the dynamic display surface size and the resolution of the medium at use. Other aspects are also important to be aware of, such as technology and document format, unlimited vertical space, interactivity, lack of typographical support (E.g. fonts can't be embedded in typical web documents) etc. [16] [36]

There are also different technological approaches to consider when laying out web document content. Should the document be processed server side or client side; the first demanding powerful hardware on traffic intensive servers, the latter relying on a slow-churning script and locally powerful hardware. Which programming environment is suitable to solving the problem, there are coarse differences of choosing a java environment based on an application server and compiled code to choosing e.g. a script-based PHP-environment. [37] [16] The choice of approach regarding design of algorithm is also an important issue based on what platform is used. There may be some layout automation algorithms that perform better in certain environments than in others.

Spatial constraints have also been researched in the perspective of potential technical utilization in already established web technologies. Badros et al developed a style sheet formalism based upon CSS (Cascading Style Sheets) to describe constraints called CCSS. In the CSS 2.0 specification there is a lack of support regarding spatial constraints (The information structure of web document defines natural constraints as a child element inherits certain constraints from it's parents and siblings). The paper shows an interesting approach to defining formalism of semantics for expressing constraints in an already existing standardized technology. [17] [22]

The research problem related to web document layout automation that has resulted in most of the publicly available research articles focuses on the problem of optimizing the layout of block elements on a limited surface. The use-cases presented in articles presenting this problem are more or less always related to newspaper presentation online. The research of Kamba et al resulted in an online newspaper called the Krakatoa Chronicle. The newspaper where based on Java Applet technology; an applet in the web browser making a server-side call for the information to present and generating a layout of the article elements on the fly. [30]

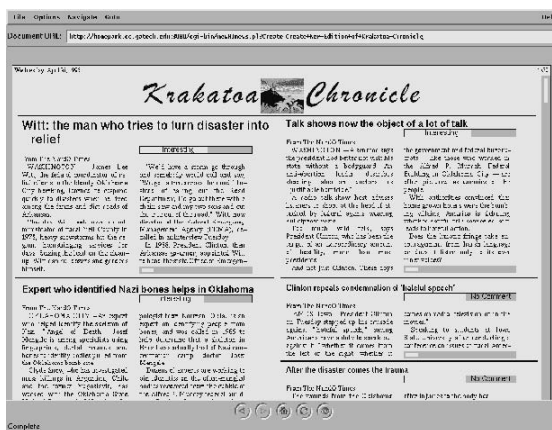


Figure 12: Kamba et al, the Krakatoa Chronicle

Gonzalez et al has a more scientific approach; they formulate the web document layout automation problem based on more general problems. They present a client-side implemented solution which tries to simulate the layout of a real newspaper for this kind of optimization. They state that previous implementations of such systems didn't generate the layouts fast enough based on the assumption that a real-time generation clearly is preferred. The solution is based on a simulated annealing algorithm for the purpose of optimizing the layout. [18] Gonzalez and Merelo have also proposed solutions to this problem in earlier research based on a server-side script-implementation of a system that optimizes the presentation of the newspaper article blocks so that the surface used is minimized. [16] The interpretation of the layout optimization problem of Gonzalez and Merelo and Gonzalez et al is derived from the general problem of bin-packing which has caught the interest of many research milieus in different areas of science.

2.2.1 Web Documents as User Interfaces

Zhang and Ren proposes a system framework called CATER for the purpose of web document layout automation. As already mentioned in section 2.1.1 this is a specialized case focusing on layout automation related to field presentation of database table columns (Relational attributes) in transactional pages. They use the term field-to-widget to formalize the mapping from database table to a form input field presentation. [15] Transactional pages refers to pages of form



Personal Information	
ID Number:	<input type="text"/>
Your Name:	<input type="text"/>
Gender:	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Others
Birthday:	<input type="text" value="1974-08-27"/> 
Birth Place:	<input type="text"/>
Education Degree:	<input type="text" value="Bachelor"/> 
Languages Speaking:	<input checked="" type="checkbox"/> English <input checked="" type="checkbox"/> Japanese <input checked="" type="checkbox"/> Chinese <input type="checkbox"/> Other
Brief Resume:	<input type="text"/>
<input type="button" value="Reset"/> <input type="button" value="Submit"/>	

Figure 13: Zhang and Ren, CATER web interface layout generator sample

interaction between user and system for information validation and persistence. The framework takes into account the processes of laying out the form field elements, data validation, persistence store, flow control and navigation. The motivation behind the CATER system is to improve the efficiency of generating transactional pages. Today the task of making the UI of transactional pages is time demanding due to the following challenges: manually building the graphic user interface widget, mapping the widget fields to the database persistence fields and complying to transactional business rules. The work is based on constraint satisfaction techniques; a constraint-based

layout policy to lay out the form fields of the graphic user interface in the respective transactional page. [15]

Lohman et al have also circulated around the problem of user interface form generation related to web document layout automation. The research of Lohman et al resulted in system for dynamic generation of context-adaptive web user interfaces. [38] They claim that the existing web engineering technologies do not consider dynamic interface modeling based on contextual knowledge. Web engineering modeling technologies that are discussed include UML-based web engineering (UWE) and the Web Modeling Language (WebML). Lohmann et al provide an ontology-driven approach to the area of context adaptive web applications. Lohmann et al describe an implemented dynamic user interfaces with XSLT-templates and CSS-stylesheets for the typical GUI (Graphical User Interface)-patterns. In addition they've used ontologies for the context adaption; a conceptual model for domain ontology and context ontologies, a navigation model for structure and composition, a view model for selection and adaption and a presentation model for GUI-characteristics. The ontologies serves as a formalism of typical modeling patterns which may be utilized in layout adaption and automation related to GUI-patterns. [38]

2.2.2 Adaptive Web Documents

Xie et al proposes a solution for adaptive web document layout in small display devices. They introduce the concept of composite documents which are scalable in both logic and structure to ensure effective information acquisition independent from the display environment. Context-dependent variable documents presented in heterogeneous environments demand dynamics in both the logical structure and graphic presentation. Xie et al discuss a novel document representation structure and an automation implementation of web documents generated from existing web sites which adapts to the display device. [26]

Goldberg et al discusses the effects of information distances in liquid layout. Based on a solution of liquid layouts that conforms the information display environment they have examined how the the display surface exploitation in both horizontal and vertical direction affect the communication effectivity. The use of liquid layouts that adapt the information presentation to the display environment and exploit the horizontal width of the display resulted in a movement accuracy with 5% to 25% time saving. [19]

2.2.3 Constraint satisfaction in web documents

Using constraints in web document layout automation has also been investigated by researchers. Badros et al proposes a technological expansion of the already existing CSS-standard to include constraint-satisfaction logic. The proposed solution was called CCSS (Constraint Cascading Style Sheet). Although the technological proposition hasn't been followed up, it represents an interesting proof-of-concept related to practical utilization of constraint-satisfaction in web document styling technologies. [28]

Hosobe claims that it is difficult to handle web documents with the already existing constraint-solving techniques. Hosobe proposes a novel algorithm called DuPlex to enable a ubiquitous solution to web document constraint solving. DuPlex is implemented to solve hybrid systems of linear constraints and one-way constraints which extends the constraint-solving library Cassowary [39] [17] As we've already described, Zhang and Ren also have proposed a solution for

web document layout automation based on the principles of constraint-satisfaction for handling form fields in transactional pages. [15]

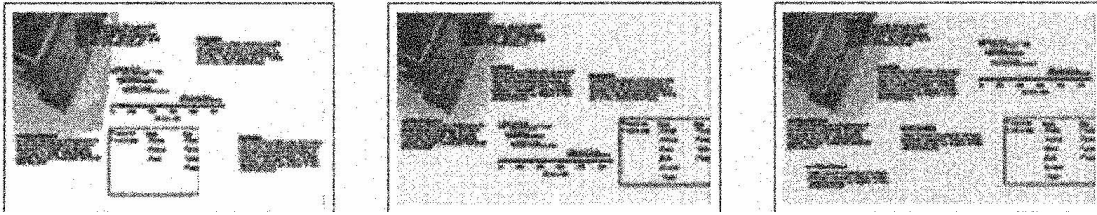


Figure 14: Bateman et al, constraints-based layout

2.3 Layout Generator Algorithms

Many algorithmical approaches to the problem of document layout automation have been implemented and tested. Approaches built upon general greedy algorithmical solutions, dynamic programming, local search algorithms, evolutionary algorithms etc.. First of all some of the techniques have been proven inefficient as the solver makes decisions bringing the algorithm to an insufficient suboptimal solution. Based on the fact that many of the problems related to layout automation often are formulated and analyzed as NP-hard, algorithms traversing the search space is deemed to land on a suboptimal solution. This doesn't mean that the algorithm needs to be satisfied with an insufficient suboptimal solutions; this suboptimal solution should be as close to the optimal solution as possible. A typical efficient approach is top-down, which suffers from higher computation time, but reaches the optimal solution based on the preferences set in the system. [4] Evolutionary and annealing algorithms has also been implemented to simulate a more biological (Genetic algorithms - GA) and physical (Simulated annealing algorithms - SA) approach to the problem. These solutions appear to be very interesting in the scope of this problem as they are more robust and flexible local search solutions.[20] The important issue regarding the choice of algorithm is making it perform in relation to the system design, depending on grammar or lingo of constraint definitions and the implementation of metrics that are useful. [40] [16] [4]

Both local (bottom-up) and global (top-down) optimization techniques have been tried for the problem of layout automation. Simon Lok and Steven Feiner's automated layout survey discuss the pro's and con's of the use of both local and global techniques. As the problems of layout automation, however they are formulated, often are of such complex nature that non-randomized algorithms are insufficient and unsuitable for the task of solving them, the use of randomized algorithms (Especially GA and SA) and heuristics are over-represented in earlier research. [4]

2.3.1 Simulated Annealing

Local search algorithms are especially suitable for dealing with computationally intractable problems; e.g. NP-hard problems unsolvable in $O(n^2)$. Local search algorithms can be generalized as algorithms that explores the solution space by sequentially moving to a nearby solution. They are very practical for many intractable problems and utilizable in different algorithmical scopes involving large solution spaces. The disadvantages are that these kinds of algorithms are unpre-

dictable and hard to analyze regarding the quality of the solution. [20] Simulated annealing are part of an physical annealing modeling paradigm of algorithms. Annealing is a term used for the gradual temperature cooling of a material to reach a crystalline state. The simulation of this physical phenomenon has been proven as an efficient model for dealing with complex problems effectively. [20] [10] [31]

Annealing algorithms have also been widely implemented in research projects related to layout automation. Penalver and Merelo claims that simulated annealing algorithms performs slightly more efficient than genetic algorithms for problems of this kind. [16] Ahmad [10] claims that simulated annealing is very effective in solving general layout design problems because of the problems complex nature. Annealing algorithms are based on a modeling of physical phenomena related to crystallization and principles of statistical mechanics. Simulated annealing is more precisely an algorithm that models the physical temperature cooling of a material to a crystalline state. The temperature determines the probability of changing to a new state at any step of the algorithm process, thus the algorithm cools down until equilibrium is reached. A state at which the algorithm have found a reasonable suboptimal solution. [20] [10]

Fogarty and Hudson have implemented a simulated annealing algorithm in their layout research framework GADGET. The system utilizes the algorithm in an evaluation optimization context for typical layout problems and provides a framework for exploring layout optimization problems in experimental settings. [31]

Another example of utilization of the simulated annealing algorithm in layout automation related problems is presented by Johari et al. They use the algorithm for optimization yellow pages pagination due to it's efficiency related to large search spaces. The layout of the massive information loaded yellow pages represent a typical intractable search space effectively handled by a simulated annealing algorithm. [29]

2.3.2 Genetic Algorithms

The genetic algorithms are part of the evolutionary paradigm of machine learning. Evolutionary modeling provides powerful tools for “fitness” simulation and evolutionary development within a framework of emergent models. Such modeling simulates the natural selection of individual solutions within a population of solutions, where each iteration strengthens the genes (E.g. attributes in layout automation) of the current generation of individuals. [41] The biological metaphor the genetic algorithms are based on makes it suitable for complex problems such as problems related to layout automation. [10] [3] [18]

Both Purvis et al and Merelo and Penalver suggest the use of a genetic algorithm for the purpose of automation in personalized documents. Purvis et al implemented an algorithm where the document is modeled as a genome with different affiliated genes denoting attributes of the genome. The attributes were weighted with priority and optimization is approached by the individual objective function scores based on the weighted sum of the attributes. [3] [16] [20] [41]

2.3.3 Greedy algorithms

Algorithms implemented under the umbrella of the term greedy may differ in structure as this term isn't precisely defined [20]. The conception of an greedy algorithm is an algorithm that

assembles the construction based on optimal step by step solution throughout the algorithm process. The algorithm determines an optimum based on predefined criteria from which it reaches the current process stage optimal solution. These types of algorithms is superior at finding globally optimal solutions for problems which could be dismantled to local rule-based subproblems. [20] [10]

In layout automation the principles of greedy algorithms have been used by several research projects, usually as a supplement or refined tuner upon other implemented algorithms. An example of this kind of implementation is the web document layout generator of Penalver and Merelo; a web client layout generator implemented with a genetic algorithm. When the genetic algorithm have done it's work the greedy algorithm takes over and further improves the generated layout. Improvements are for example minimization of empty space between content containers and other problems that evolutionary algorithms as the genetic algorithms may not perfectionate as it merely reaches an suboptimal solution. [16]

Xing Xie et al suggests the use of a greedy algorithm for web layout screen adaption. They formulate the web layout problem as a variance of the 0-1 knapsack problem and thereby proposes it solved by this type of algorithm. Xie et al proposes a solution to map existing information content presented as a web document to a presentation adapted to a different viewing environment (E.g. lower resolution screen/web client window). [26] The term knapsack problem refers to the problem of filling a knapsack of capacity W of items with a given weight w_i and value v_i , so that the total value in the knapsack is maximized without the total weight passing the capacity of the knapsack. In the 0-1 knapsack problem number of instantiations of each item are restricted to maximum one. Xie et al claims that when the web layout problem is formulated as a variance of the knapsack problem, the use of a greedy algorithm efficiently solves these kind of problems. Their implementation of the greedy algorithm tries to add information blocks to the block set one by one in decreasing order. If there is no possible way to add a block, it won't be further considered. Xie et al mentions that this approach have been proven effective in traditional knapsack problems. [26]

2.3.4 Heuristics

Heuristic methods are based on instructed machine evaluated guessing; a trial-and-error methodology often used for complex and uncertain problems with no obvious algorithmical solution. The trial-and-error paradigms are based on instructed rules of discovery and invention. [41] Heuristics can be formalized as the use of rule-based space search for choosing candidates (The term branches is used for the choice of "path") which leads to a sufficient and acceptable problem solution. [41] As general layout automation problems (E.g. bin-packing, knapsack problem) often are NP-hard problems, researchers have often turned to heuristics as it provides effective means for solving such problems. [10]

Kroener uses heuristics for problem identification regarding layout generation in large search spaces. The layout generator compares the current state with known patterns of layout flaws and if a pattern is identified the layout generator tries to identify the object responsible and modify this object such that the next state resolves from the pattern. The approach is presented as a heuristical mapping from content to content presentation. [35]

Kim et al have suggested to use a heuristical approach for space efficiency within layout. This is a well known problem in the field of VLSI and are relevant in document layout automation as the white-space efficiency on the display medium is an important attribute related to layout quality. [11] The approach involves a new heuristic sliding technique which finds a suboptimal layout state without overlapping between elements. The system is able to assess a reasonable geometrical sliding direction of elements at state so that there will be no element intersection. [40]

2.4 Preference Attributes in Layout Generators

Document layout generators are dependent on some kind of evaluation of the characteristics layout to be able to optimize the layout quality. The characteristics of a layout can be formulated as attributes that affect the overall layout quality. These attributes are can be preferences as balance, alignment, symmetry etc. Just about every research project involving an algorithmical evaluation based approach to document layout automation utilizes the quantification of certain preference attributes for the purpose of optimizing the layout regarding quality.

2.4.1 General Preference Attributes

Many research projects have focused on the general features of document layout. What distinguishes good layout from bad layout? Which characteristics are present in qualitatively good layout and how are they measured? Ngo has made a major effort in defining preference attributes. His research on the area has resulted in a both a robust preference attribute set, a possible weighting of the preference attributes effect on the overall layout. The following list describes the preference attributes as defined by Ngo: [5]

- Balance - The computed difference between total weighting of elements on each side of the horizontal and vertical axis.
- Equilibrium - The computed difference between the center of mass and the physical center of the presentation medium.
- Symmetry - The extent to which the screen is symmetrical in the directions vertical, horizontal and diagonal.
- Sequence - A measure of how information is ordered in hierarchy of perceptual prominence corresponding to the reading sequence.
- Cohesion - The extent to which the elements have the same aspect ratio.
- Unity - The extent to which elements visually belong together.
- Proportion - The comparative relation between the dimension of the elements and certain proportional shapes.
- Simplicity - The extent to which elements parts are visually minimized.
- Density - The extent to which the percentage of element areas on the presentation area is equal to the optimal level.
- Regularity - The extent to which the alignment points are evenly spaced.

- Economy - The extent to which the elements are similar in size.
- Homogeneity - A measure of how evenly the elements are distributed among quadrants.
- Rhythm - The extent to which the elements are systematically ordered.

Ngo discuss the attribute set on the basis of two experiments around screen design aesthetics. The experiments were used to develop and validate a system for predicting user acceptance based on screen characteristics.

Harrington et al have also worked on aesthetics measures of document layout. They describe metrics for measure of characteristics / attributes related to document layout. In addition to formulations of the metrics for quantifying these attributes, Harrington et al provide an interesting overall quality quantifying metric. The following preference attributes are defined by Harrington et al: [11]

- Alignment - Measure for how well element edges align.
- Regularity - Measure for how well alignment positions are spaced in a regular fashion.
- Uniform separation - Measure for how well spacing between elements is homogeneous.
- Balance - Harrington et al stress that this is a major factor of aesthetics in layout. They describe two ways of defining balance: *Centered balance* is the balance of the elements related to the visual weight distributed from the visual center of a page (This is what Lok et al defines as crystallographic balance [9]) and *left-right balance* which is the vertical distribution of weight.
- White-space fraction - The white space should total about half the total page area.
- White-space free-flow - Measure for how well the white-space is connected to the margins. (Large white space areas surrounded by elements in the middle of a layout is called “trapped” white space and degrades the overall quality of a layout)
- Proportion - Measure of how well the elements are proportioned in relation to each other compared to certain pleasing ratios (E.g the “golden ratio”; $R = \frac{2}{1+\sqrt{5}}$)
- Uniformity - Measure of how well elements are uniformly distributed over a page.
- Page security - Measure of how the elements are positioned “securely” within the boundaries of the page. Elements near edges of the page might appear insecure and could trig the feeling of information “falling off”.

Bauerly and Liu also discusses preference attributes and characteristics of document layout in a broad perspective. They focus on the operationalizing how the quantity of layout elements and the attributes symmetry and balance affect the quality of layout. Experiments conducted in their research indicate that expert subjects are highly proficient in judging symmetry and balance in both horizontal and vertical directions. This justifies the quantification methodology implemented in their research. [6]

2.4.2 Balance

Balance is an important attribute of document layout and have major effect on the perceived visual quality which is widely documented. [11] [6] [9] Simon Lok et al describes three different types of visual balance; symmetric balance, asymmetric balance, radial balance and crystallographic balance. Symmetric balance is the notion of mirrored symmetry of balanced elements across the vertical and horizontal axis. Asymmetric is the lack of symmetry, but still the presence of balance in layout.



Figure 15: Lok and Feiner; a sample layout generated by layout automation component BalanceManager

Radial balance is the notion of balance around a single point. Crystallographic balance is the opposite and rather an overall balance in both the vertical and horizontal directions. Lok et al suggests an implementation of a crystallographic balance operator based on the concept of visual weight; the notion of layout elements' distributed visual weight on the plane document surface (Fig. 15). Lok et al suggests to implement the balance operator with a visual weight map calculated from the luminance of the elements layed out. An area of high luminance weights higher than one with low luminance. Luminance unevenly distributed in the vertical or horizontal direction is disfavored in the perspective of evaluating the balance operator (Fig. 21). [9]

Bauerly and Liu also presents a balance operator in an layout evaluation perspective. They define a balance point as the Cartesian coordinate of the center of layout visual mass. The center of mass can be found by analyzing global pixel visual weight distribution. They suggest a pixel by pixel analysis metric as an operationalization of balance for calculating both the vertical and horizontal balance. [6]

Harrington et al describes balance as *a major factor in layout aesthetics*. They define two types of balance; centered balance and left-right balance. The definition of centered balance as presented by Harrington et al resembles the definition of crystallographic balance as termed by Lok and Feiner; the balanced distribution of layout objects from the visual center of the display surface. [11] [4] Harrington proposes a pure mathematical description of how to operationalize the balance attribute in document layout.

Ahmad also defines balance in a similar manner as Bauerly and Liu and Lok and Feiner; as the difference between total visual weight of layout objects on each side of the vertical and horizontal axes. Ahmad presents a mathematical metric for measuring the weight distribution and balance of a layout. [10]

2.4.3 Equilibrium

Equilibrium is a layout characteristic which is quite hard to grasp. Ngo defines the operationalization of equilibrium as *computed as the difference between the centre of mass of the displayed components and the physical centre of the screen*. [5]

2.4.4 Cohesion

Ahmad defines cohesion as *the extent to which modules (blocks) on each side of vertical and horizontal axes of a layout configuration have same aspect ratio*. [10]

2.4.5 Proportion

Proportionality is a generally well-known geometrical and mathematical term. Harrington et al claims that some proportions are more aesthetically pleasing than others; e.g. the “golden ratio”, widely utilized in architecture, arts and visual design. They present a mathematical approach to operationalizing proportion in layout evaluation based on certain ratios. [11]

2.4.6 Density

Ahmad describes density as uniform distribution of layout objects on the entire surface available. Ahmad presents an operationalization for measuring the distribution of objects in the four quadrants of a presentation surface. Uniform layout object distribution in these quadrants determines a good density score. [10] Ngo defines density as *the extent to which the percentage of component areas on the entire screen is equal to the optimal level*. [5]

2.4.7 Alignment

The alignment operator is widely implemented as a layout support tool in the majority of software adopted by the design industry. Vendors as Adobe and Quark have provided several tools for automating graphical object geometrical relationships including the alignment tool. Harrington et al presents alignment as an important factor in measuring layout aesthetics. The alignment of objects based on certain edges of the object immediately ensures a more consistent and homogeneous layout, and thereby a more appealing layout.

Alignment means to arrange the graphical objects by one or more of these edge points, either in the vertical direction or horizontal direction. E.g. aligning two objects by their left edge position, or even better, aligning by their left and right edge position usually enhances the general layout quality. Harrington et al presents an histogram-based approach to operationalizing and evaluating alignment; the graphical objects is placed in a histogram bin based on their position (One histogram for left edge point, one for top edge point etc.). Neighboring edge points in both the vertical and horizontal direction is compared regarding alignment based on the histogram bin they are placed. Histogram-bins of high clustering determines a well aligned layout (E.g. having all object edge points accumulated in only two bins means a good alignment score). [11]

2.4.8 Regularity

Harrington et al describes regularity as the placement of objects in a regular fashion. This means distribution of layout objects by edge points in a regular spatial interval. They present operationalization of regularity as an algorithm similar to that of alignment; using a histogram-based approach to measure the regularity score as described above. [11]

Ahmad rather describes distribution / count than regularity. The terms are more or less the same, both related to the equal distribution of layout objects on the surface. Ahmad defines distribution as *the extent to which modules (objects) are equally divided, or distributed, in a layout design*. Ahmad also presents a mathematical operationalization of distribution described later in this chapter. [10]

Ngo describes regularity as an attribute based on the principle of alignment; *Regularity is the extent to which the alignment points are consistently spaced*. [5]

2.4.9 Uniform separation

Harrington et al also implements the histogram-based approach for measuring uniform separation between layout objects. Uniform separation and regularity are both related to the layout object positions. The difference is that uniform separation of objects means that objects are separated in a consistent manner; e.g. the separation between bottom edge point of one object and the top edge point of another is uniform for all vertically neighboring objects. [11]

2.4.10 Uniformity

Harrington et al describes uniformity as uniform distribution of layout objects on the presentation surface. This is generally preferred aesthetically. Harrington et al defines non-uniformity as the variance of visual density. The uniformity score is low if the visual density distribution highly varies on the presentation surface. [11]

2.4.11 Symmetry

Document layout symmetry is the geometrical correspondance of graphical objects on each side of the vertical, horizontal and diagonal axes. [5] [6] Bauerly and Liu proposes an pixel-by-pixel analysis as an approach to measuring symmetry in document layout. The concept involves calculating the visual distance of two adjacent pixels on each side of the axis. A minimized sum of visual distances between the pixels determines a better symmetry score. [6] This principle is derived from the visually perceptive concept of symmetry which is further derived from the geometrical and mathematical concept of symmetry; the geometric reflection of geometrical objects on both sides of a line.

2.4.12 Economy

The term economy related to document layout is loosely based on the ubiquitous term resource economy. The resource in the perspective of document layout is the surface available by the document format properties. Ngo defines economy as *the extent to which the components are similar in size*. Rectangular elements of the same size implies easier optimization of surface resource efficiency. [5]

2.4.13 Homogeneity

Oliveira describes the notion of homogeneous page cover as the evenly placement and distribution of layout elements. Elements should be evenly placed, spaced and distributed and not clutter into one region of the document. [21] Ngo defines the term homogeneity as *a measure of how evenly the components are distributed among the quadrants*. [5] Aesthetically a homogeneous distribution of layout elements is preferred rather than a somewhat random distribution;

the notion of homogeneity is closely related to and determined by the properties of other layout characteristics such as symmetry, alignment, uniformity and regularity. [21]

2.4.14 White-space Optimization

White-space optimization relates to the problem of reducing the white-space in a document layout. The problem of optimizing this preference attribute is in its purest form a direct derivation from the bin-packing problem.

Harrington et al presents two operators related to white space optimization. They define white-space fraction as the total white space of the information plane including margins. This should total about half of the total page area. In addition they operationalize the attribute white-space free-flow. White-space free-flow is measured on how well the white space is connected to the margins. “Trapped” white space, the notion of white-space trapped between surrounding elements with no connection to the margins, has a negative effect on this operator. [11]

Oliveira presents two algorithmical approaches to the problem of white-space optimization. Rather than focusing on the general area of document layout automation, Oliveira dives deeper in to the special problem of layout automation related to chronological information. [21] In this specialized version of the document layout automation problem Oliveira claims that the bin-packing problem is non-utilizable because of additional requirements to the information organization. These requirements are the demand for homogeneous presentation of the information and the information in reading order

2.5 Operationalizing Layout Quality

2.5.1 Operationalizing Attributes

The quantification of preference attributes is essential in the perspective of operationalizing layout quality. This operationalization involves defining metric for attribute evaluation, measuring the attributes and defining a metric for calculating the overall layout quality based on the attribute evaluations.

Harrington et al have proposed a set of preference attributes related to layout quality and defined metrics for calculating these attributes. These metrics are important in the process of operationalizing layout quality and utilizing preference attributes in the perspective of layout automation. They present a non-linear combination of heuristic measures of attributes that degrade the aesthetic quality of a layout: [11]

- Heuristic measures for a set of features
- Overall (Explained in equation 2.24)

$$V = \left(\sum w_i (d + V_i)^{-p} \right)^{-1/p} - d \quad (2.1)$$

- Alignment (Distance z)

$$V = A/(A + z) \quad (2.2)$$

- Regularity: Distance between corresponding edges of neighbors (Might be expensive). An alternative is to examine the distance between alignment positions.
- Separation: Distance between vertical or horizontal midpoint of two objects

- Balance: Right-left balance and centered balance.
- White-space fraction: Total about half the total page area (Design rule)

$$V = 1 - 4 \left(\left(\sum A_i/A_p \right) - 0.5 \right)^2 \quad (2.3)$$

- White-space free flow: A sum of all points that lie between a page edge and the corresponding profile boundary is computed. The area of the content is added and the difference from the page area is determined.
- Proportion: Golden ratio R as ideal

$$V = 1 - |Z_i - R|/R \quad (2.4)$$

- Uniformity: Variance of the visual density (Visual weight divided by its area)

$$D_i = M_i/A_i \quad (2.5)$$

Average:

$$D_i = \sum M_i/A_{pi} \quad (2.6)$$

Due to difficulties of defining and measuring aesthetic pleasantness, the article of Alex Faria and de Oliveira suggests to avoid measuring aesthetics as an intrinsic property of an instance, rather consider aesthetics as the conservation of the properties of the original template (Relative aesthetics). [8] Measures should be defined by continuous functions. The functions of Faria and de Oliveira are highly influenced by the work of Harrington et al.

- Overall aesthetic score V , i is the index of the aesthetic measure (Explained in equation 2.24):

$$V = \left(\sum w_i (d + V_i)^{-p} \right)^{-1/p} - d \quad (2.7)$$

- Color score as RGB distance from original:

$$V = 1 - \frac{1}{\sqrt{1/3}} \sqrt{(R_0 - R_i) + (G_0 - G_i) + (B_0 - B_i)} \quad (2.8)$$

- Page cover (White space): The ideal ratio is 50%. Area of page is denoted A_p and area of content is denoted A_i .

$$V = 1 - \left(\left| \sum A_i/A_p - 0.5 \right| * 2 \right) \quad (2.9)$$

- Fonts: Suggests table based penalty relation between font families and absolute difference of font size
- Horizontal and vertical distortion (Resembles resizing operator)

$$V = 1 - (|h_{orig} - h_{dist}|/h_p) \quad (2.10)$$

- Horizontal position and vertical position (Resembles balance operator)

$$V = 1 - (|X_{orig} - X_{dist}|/w_p) \quad (2.11)$$

- Distance to neighbor

$$V = 1 - (|h_{orig} - h_{dist}| / h_p) \quad (2.12)$$

Ngo also provide some knowledge to operationalize the preference attributes in document layout. Although he doesn't provide any mathematical metrics for utilization in measuring these attributes, he describes the very nature of each attributes so that possible metrics could be derived from the description.

Ahmad presents metrics for certain layout attributes. He states that understanding these attributes are the key to a functional and practical operationalization of them in an automation perspective and describes this automation as an intricate undertaking. Ahmad presents mathematical metrics and operationalization of four specific layout attributes; cohesion, balance, distribution / count and density. The operators presented have certain similarities with operators presented by Harrington et al and Bauerly and Liu.

- Cohesion (Aspect ratio AR difference in the layout objects of the pairwise four quadrants of a layout)

$$\text{cohesion_D} = \sum_{k=1}^4 \max AR_k - \min AR_k \quad (2.13)$$

Total deviation cohesion_D_total is found by using the sum of root mean square values of the pairwise quadrants (This measure is also presented normalized):

$$\text{cohesion_D_total} = \sqrt{\sum_{k=1}^3 \sum_{l=k+1}^4 (\text{cohesion_D}_k - \text{cohesion_D}_l)^2} \quad (2.14)$$

- Balance is operationalized as the difference between total area of layout objects in the pairwise comparison of quadrants:

$$\text{balance_D}_k = \max \text{Area}_k - \min \text{Area}_k \quad (2.15)$$

Overall balance based on the pairwise difference:

$$\text{balance_D} = \sum_{k=1}^4 (\max \text{Area}_k - \min \text{Area}_k)^2 \quad (2.16)$$

- Distribution / count is operationalized as the difference between total number of layout objects Count_k and Count_l in the pairwise quadrants:

$$\text{Count_D}_{k,l} = \text{Count}_k - \text{Count}_l \quad (2.17)$$

Total distribution score (This measure is also presented normalized):

$$\text{Count_D_total} = \sqrt{\sum_{k=1}^3 \sum_{l=k+1}^4 (\text{Count}_k - \text{Count}_l)^2} \quad (2.18)$$

- Density is operationalized as the pairwise difference the sum of layout object areas in a quadrant k and a quadrant l :

$$\text{Density_D}_{k,l} = \text{AQ}_k - \text{AQ}_l \quad (2.19)$$

Total density score (This measure is also presented normalized):

$$\text{Density_D_total} = \sqrt{\sum_{k=1}^3 \sum_{l=k+1}^4 (\text{AQ}_k - \text{AQ}_l)^2} \quad (2.20)$$

Bauerly and Liu provides useful metrics for layout attribute operationalization related to several document layout characteristics. The metrics presented in the articles mainly focuses on pixel analysis (Fig. 16) of layouts in an aesthetic perspective.

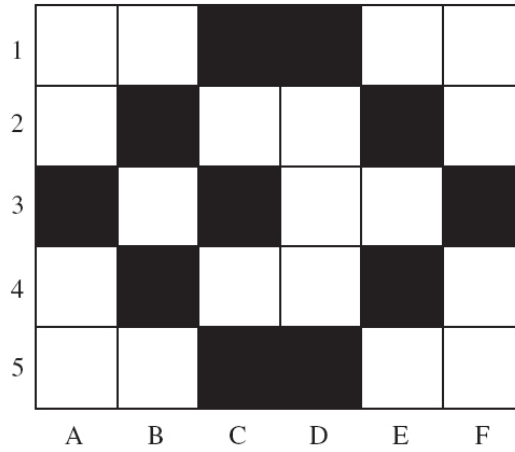


Figure 16: Bauerly and Liu pixel-based attribute analysis (Symmetry). Perfect symmetry score except for comparing the pixels C3 and D3

- Symmetry, where n is the number of comparisons, m is the perpendicular length to the reflection line.

$$s = \frac{2}{3nm} \sum_{j=1}^n \sum_{i=1}^m X_{ij} \left(1 + \frac{j-1}{n-1}\right) \quad (2.21)$$

- Balance (Unweighted pixel analysis), where (x_b, y_b) is the centre of mass point from where to calculate the balance. (Two metrics for respectively vertical and horizontal balance)

$$\sum_{x=1}^w W_x (x - x_b) = 0 \quad (2.22)$$

$$\sum_{y=1}^w W_y (y - y_b) = 0 \quad (2.23)$$

2.5.2 Operationalizing Overall Layout Quality

The overall evaluation of a layout is a function of the individual attributes evaluation. Although there have been several research projects related to operationalizing specific layout attributes there is a lack of thorough research literature related to the overall evaluation of a layout. It seems as though the research projects utilize relatively subjective attribute weighting schemes of the format $O = w_1 e_1 + w_2 e_2 + \dots + w_n e_n$, where w_i denotes the weight (Usually $w_i \in \{0, 1\}$) and e_i denotes the evaluation of a specific attribute (Usually $e_i \in \{0, 1\}$).

A first step to develop more robust evaluation techniques based on attribute evaluation is clearly to enhance the overall quality measure metrics. Ngo has taken it a step further by exploring how the different preference attributes are favorable and in what degree they are favorable. Ngo presents interesting results related to how the professionals actually are weighting the characteristics of layout, deliberately or not. Ngo also presents a regression analysis of attribute evaluation in relation to overall quality evaluation. This analysis indicate that there are huge differences in how the separate attributes are affecting the overall quality of a certain layout. [5]

Harrington et al proposes a metric for the overall evaluation of layout quality based on separate attributes evaluation. This metric takes into account the fact that one bad attribute contributor can degrade the overall quality of the layout regardless of the evaluation of the other attributes. [11]

$$V = \left(\sum w_i (d + V_i)^{-p} \right)^{-1/p} - d \quad (2.24)$$

V denotes the overall quality evaluation, w_i the attribute weights, V_i the evaluation score of the attributes, p introduces a non-linearity and d is slightly larger than 0 (To avoid dividing by zero). The evaluation scores are values between 0 and 1(good). This equation ensures that specific attrutes with especially low evaluation scores have a huge impact on the total sum of the evaluation.

3 Methodology

3.1 Synopsis

The study was planned and carried out based on the research problem and research questions formulated in the introduction. In the project planning stage the literature study early showed a lack for publicly available testing tools for research problems of this kind. Research question 1 demanded a prototype testing system to facilitate an environment to study the phenomena of document layout quality. The nature of the research question demands an operationalization of both document layout attributes and document layout quality. In addition participants from the design industry were needed to evaluate the phenomenon document layout quality, so that we were able to measure perceived quality of a layout generator with only a balance operator implemented compared to one implemented with an extended set of attribute operators. The best suited participants for measuring the notion of quality in document layout should be persons associated with the profession of visual communication. Designers, typographers and others employed in the design industry are trained to evaluate and create graphical layout. The professionals used in this study were employed and educated in professions as design, art direction, typography, media design and web design. Some of the participants were professionally active in several areas of design, others were only focusing on a particular area (E.g web design).

The study is implemented as both an interview of the participants and as a survey where the participants ranked document layouts on different design criteria. These design criteria were preference attributes of document layout which the participants were to evaluate based on their layout knowledge and experience. The attributes evaluated were both the LaG-system implemented document layout attributes balance, alignment and equilibrium in addition to other attributes of layout. The survey was modeled to both pick up how the implemented attributes (Balance, alignment and equilibrium) and the overall quality of the layout was affected by the enhanced attribute set in the prototype system.

A grounded theory study of the research literature where conducted to establish knowledge related to research question 2. The proposed architecture and the general components of the modeling framework were derived from the related literature. This modeling framework was practically used to model the document format-independent layout automation system LaG. The experiences from the practical system modeling resulted in a further development of the modeling framework to ensure layout automation system modeling generalization.

3.1.1 Study Objective

There are two main objectives of the study, both related to the research questions of this thesis. Research question 1 demands an operationalization of both document layout attributes and overall document layout quality. The overall quality of a layout needs to be measured both by machine and human to further understand if an extended set of attribute operators helps this quality. Thus the study objective related to the first research question is to measure visual quality of a document layout in both the perspective of human perception and the system evaluation.

The main purpose of establishing a system modeling framework was to provide a general framework for modeling a wide range of layout automation systems of diverse application. By providing this modeling framework the workload of modeling layout automation systems based on evaluation techniques can be reduced.

3.2 Participants and Sampling

The study implemented in this thesis involves measuring qualitative attributes of visual aesthetics. It is quite difficult to grasp the abstract notion of quality in graphical design. Graphical design is closely related to arts, which both are crafts highly dependent on human cognitive intuition and visual perception skills. These skills aren't learned by merely observing a couple of designed posters, but demands years of both education and professional experience. We sought to find this experience in two Norwegian advertising and web companies situated in the town Gjøvik.

Due to the complexity of the problems related to evaluating the notion of quality in document layout automation it is important to choose a robust set of samples. This thesis focuses on establishing knowledge of the effect different preference attribute operators have on each other in document layout automation systems. This means that we needed samples generated from systems with all the respective attribute evaluators implemented. The purpose of this study was to find inter-element influence on the balance, alignment and equilibrium attribute operators; the important aspect is the qualitative difference / distance between using only the balance operator compared to using additional operators. It is interesting to investigate how the specific attribute operators influence each other and the overall quality of a generated layout.

3.2.1 Participants

The choice of participants is important in this kind study as we are dependent on getting professional qualitative evaluations of aesthetical characteristics. The participants in the study were chosen based on their education and professional experience. We could have used designer students from colleges and universities, but the practitioners eye for layout characteristics is better trained in the direction of de facto industry design standards in addition to retaining design study education. Other studies related to problems of document layout automation have been implemented with expert participants in the same manner this project has.

David Ngo conducted a study involving seven GUI designers from a multimedia company for the evaluation of layouts in the perspective of specific layout attributes to establish knowledge related to general document layout characteristics. [5] Cie et al used 21 university students as respondents in a study of how web site aesthetics influenced consumer behavior online. [25] Faria and de Oliveira performed layout evaluation experiments with 6 participating graphic artists and 9 students of computer science. The participants were to rank documents according to its aesthetic value. [8] De Angeli et al performed studies on aesthetics of web site design patterns involving 28 undergraduate students (Described as experts) with basic knowledge of HCI. The participants were to evaluate two web sites produced based on differing design criteria. [42]

3.2.2 Sampling

The survey presented in this thesis where dependent on a proper sampling of document layouts for the use in human evaluation. A robust set of samples ensures a strengthened validity in the evaluation results collected in the survey.

Practical implementations of surveys related to the problem area presented in this thesis have a wide variety of both sampling design and sampling size. David Ngo presents a mapping on how different characteristics of document layout affects the general aesthetic of the design. The set

of samples used consists of 57 screens from a variety of multimedia systems. Ngo claims that a stratified sampling related to predefined characteristics is virtually impossible due to the interrelated nature of these characteristics. He also concludes that the limited population samples used in the study preclude a possible generalization. [5] The conclusions of Ngo related to sampling is in broad contrast to the assessments of de Angeli et al; the study presented in the research article implements a sampling size of two document layouts. De Angeli et al presents a study of how information quality is affected by interaction style implemented in the user interface. In practice the document layout samples used were two versions of an existing web site tweaked to fit the characteristics of different interaction styles. [42] The work of Cai et al related to consumer online behavior uses a sampling design with 16 website interface samples presented to the participants in the laboratory experiment. The participants were to evaluate aesthetic aspects of the individual designs. Cai et al claims that the study implemented in this study is one of the earliest experimental attempts to manipulate characteristic aspects of aesthetics related to web documents. [25]

The population of distinct document layouts of a graphical presentation consisting of a set of graphical elements is almost impossible to measure. The sampling design used in this research project is based on random sampling of two defined strata; document layouts generated from a generator implemented with a balance operator and document layouts generated from a generator implemented with balance, alignment and equilibrium operators. By using stratified sampling of generated layouts we ensure an equal representation of the samples of both generator modes. [43]

All the samples were persisted in database storage for use in the participant survey. To avoid that the document layout evaluations suffered from participant fatigue during the survey, the system presents the generated documents in a random order. In this way the document layout average evaluation shouldn't be affected of participants calibrating to the sample characteristics during the presentation.

3.2.3 Rationale

The qualitative aspects of the study implemented in these kind of research projects is a sampling design challenge. A characteristics evaluation of document layout implies that the participant assessing these characteristics possesses the skills, experience and knowledge related to the area of design and graphical aesthetics. This ensures evaluation integrity and correctness, which more or less determines the robustness and external validity of the collected data in the study. [43] As already stated we use a stratified randomized sampling of the document layout population. The rationale for using modes related to the strata is the formulation of research question 1; revealing the affect an extended evaluation attribute set has on the overall generated document layout quality.

The participants were set to evaluate a sample set of document layout generated from the LaG-system. The sample set included a total of 50 document layout samples. As mentioned the system supports two modes; a basic mode only with a balance attribute evaluator active and an extended mode with all three attribute evaluators active (Balance, alignment and equilibrium). There were used 15 samples of the basic mode and 10 of the extended mode, which should be

sufficient sampling set size for this study based on experiences drawn from similar research. [5] [25] The participants were presented the samples one by one in a random manner without any indication of which stratum the particular sample was a part of. This further ensures the external validity of the study data as certain experiment-related participant effects, such as the Hawthorne effect and the Halo effect, are minimized. The selection of generated document layout samples were made in one sequence of 25 generation iterations for each system mode, which in practice is a randomized selection of samples.

The generated document layouts are all layouts of web documents. The web document layouts are presented on a limited surface constrained by the height and width of a 1024 pixel by 1024 pixel XHTML-DIV container. The project aims to establish knowledge related to both general document layout (Thereby the constrained surface) and technical principles of web document layout generation (Thereby the use of XHTML as format).

3.3 Survey implementation

The main aspects of the formulated question is the operationalizing of document layout quality, operationalizing document layout attributes and implement testing tools that provide generated layouts both with and without an extended set of attribute operators. In addition these aspects needs to be quantified in a manner that makes them measurable.

The survey consists of two parts; design practitioners evaluating a set of generated web document layouts and a design practitioner interview related to the utilization of document layout automation. The first part serves as a tool to help answering the problem aspects of research question 1. The second part is implemented to gather additional information about how the practitioner evaluate the current use of and demand for document layout automation tools and systems within the design industry. This is interesting in perspective of mapping the practitioners comprehension related to the research field and useful in the perspective of practitioner awareness as participants in the study.

The interview was conducted preceding the main survey to trigger this awareness. Both the main survey and the interview are thoroughly described in the following sections. The complete session of interview and survey lasted about 45-60 minutes for each participant. The full implementation of the survey was evidently very time consuming and it was hard for the designers to find time for carrying out the survey in a strained work day. Although we would have wanted to conduct the survey on additional participants the total amount was limited to 6 due to the scarce time resource of the participating design companies and participants. Still this should be sufficient based on the layout sample size used and according to similar study implementations from earlier research. [5] [8]

3.3.1 Practitioner Interview

Simple techniques of document layout automation are already being used in the design industry. Professional document layout practitioners as graphical designers are seldom aware of the tools they already utilize to efficiate tasks related to their type of work. Design production software and system vendors continuously compete in implementing features for increasing the efficiency in the workflow of the design process. Practitioners imperceptibly adapts to the software evolu-

tion by incorporating the use of such tools in their workflow whenever additional features are introduced.

The study presented in this thesis provides an practitioner interview related to the use of automation tools and the demand for automation tools in this perspective. The main purpose of the interview is to establish knowledge about the use of layout automation tools, the demand for such tools and triggering the participant cognition related to their own practice. The first two are motivated by the lack of documentation related to practical implementation of automation tools related to document layout. The latter is motivated by strengthening the research study robustness. The interview questions serves as a catalyst for participant cognition, comprehension and awareness of aspects related to document layout. E.g. as the participant educe the latent consciousness related to the use of alignment operators in Adobe Illustrator, the theoretical principles of the particular attribute is cognitively triggered. This strengthens the integrity of evaluations made in the main survey.

Practical Implementation

Prior to the survey interview the participants signs an informed consent. The interview session starts with a presentation of where the research is related to the general problem of document layout automation. Participants involved in the survey may have high expectations to how a document layout generator performs. The technologies used for general layout automation is still immature, producing resulting layouts far from the standards of a designer. By showing the participants samples of layouts presented in related research literature, the participants' expectations should be calibrated to a reasonable level. Participants evaluating attributes based on their professional preferences may result in data which can't be properly analyzed. On the other hand the risk by providing samples of layouts prior to the survey is that it may affect the participants in such a manner that the evaluations are to liberal.

The interview is a mix of questions related to the use and demand for automation tools related to document layout and use case statements for which the use and demand for such tools exist. The questions are generally for the investigation of how practitioners utilize tools for efficiating the design process and for mapping how practitioners see the demand for further efficiency. The use cases presented in the interview is formulated in a way that forces the practitioner to make a stand related to how and where utilization of layout automation is reasonable. This part is implemented to force the participant to cognitively revolve around questions related to document layout automation.

Collecting Data

Notes from the interviews were written down and logged. The results are generally used as a basis for further literature review related to the use of document layout automation tools in the design industry.

3.3.2 Layout Survey

The designer always continuously evaluate the product he or she is working on during the design process. However the design product evolves, either as a result of primitive trial and error process steps or as theoretically grounded decisions, a step by step evaluation occurs to mold the final product. Evaluation techniques of document layout automation tries to operationalize

these process evaluation steps. The computer use evaluations based on metrics that quantify the notion of layout quality; a relatively infantile simulation of the design trial and error stepwise process. [4] Research question 1 of this thesis is problemizing the core element of these evaluations; the use of attribute evaluators that quantify the notion of specific characteristics of a document layout.

As the study focuses on establishing facts regarding the performance of existing document layout automation techniques, there is a need to analyze the collected data in relation to this problem area. The collected data is based on a ranking metric related to layout characteristics / preference attributes, makes it possible to more precisely quantify and analyze the results from the collection. Results are presented as frequencies of each layout characteristic and relating this to layout automation techniques of certain attributes (and dummy layout made by professionals) makes it possible to potentially observe a correlation between generated layout and designer preference.

The survey is the main methodological tool to gather data related to how the use of preference attribute evaluators affect the perceived quality of a generated layout. The survey is designed and implemented with the purpose of retrieving these data. The participant evaluates several generated layouts on certain layout attributes as variables. In addition the participant evaluates the overall score of the layout.

Practical Implementation

The participants were seated in a room accompanied solely by a computer. A survey version of the LaG system was running on the computer. This implementation serves as a survey interface which presents the 25 layout samples in a random order and providing a panel for the users to rank preference attributes and overall score of the particular layout. The participant ranks the layouts one by one, until the presentation session completes. Preference attributes that the participant were theoretically unsure of, were explained (based on objective theory from related literature).

Layout Evaluation

The evaluation panel consists of user interfaces for ranking eight different attributes related to document layout in addition to the overall score. The following attributes were set to be evaluated by the participants (On an interval scale from zero to ten):

- Balance - The difference between visual weighting of elements on each side of the horizontal and vertical axis. [5] [11]
- Vertical balance - The difference between visual weighting of elements on each side of the horizontal axis. [9] [11]
- Horizontal balance - The difference between visual weighting of elements on each side of the vertical axis. [9] [11]
- Symmetry - The extent to which the presentation surface is symmetrical in vertical, horizontal and diagonal direction. [5] [6]
- Alignment - The extent to which the corresponding sides of the elements are aligned. [11]

- Regularity - The extent to which the alignment points are consistently spaced. [5] [10] [11]
- Equilibrium - The difference between center of mass and physical center of presentation surface. [5] [10]
- Efficiency - The effective usage of presentation surface. [5] [11]

Interview

The questions in the interview session were formulated to give answers related to how practitioners use layout automation tools and where they see a demand for such tools. The motivation for conducting an interview was to establish the industry de facto standards for work flow in addition to investigating how they consider the possible utilization of automation tools. The interview also serves as a participant cognitive stimulation related to document layout automation; self-help to refreshing their latent knowledge related to document layout automation. The interview was divided into two parts. In the first part the participants answered questions related to how they see potential improve in efficiency related to their work tasks in the process of composing layout. The second part focused on how they already use automation tools. The following interview questions and statements were formulated (The statements were generally followed up by questions related to the comment of the participant):

First part of the interview. “Efficient work tasks in the process of composing layout. Comment the statements”¹

1. I often work with graphical design, where several graphical elements are laid out in a relatively repetitive manner.
2. Design tasks that I work with is highly creative of nature.
3. I need automation tools that eases tedious and repetitive tasks in the process of composing layout (Consider this statements related to the examples below)
 1. General grid-based layout.
 2. Layout of the front of a novel in book form.
 3. Layout of articles on the main page of an online newspaper.
 4. Layout of a music festival poster in a marketing context.
 5. Layout of a search result list of books done in an online book store.
 6. Layout of a large number of advertisements in a printed publication.

Second part of the interview. “Use of well-established tools in the process of composing layout. Comment the statements”

1. I use general tools (Grid-systems, templates etc.) for easing layout composition in my profession.

¹Interview is available in appendix D

2. I use automation tools for layout composition in my profession.
3. If you agree to the previous statement; which tools do you use to automate the process of composing layout.

Collecting Data

The evaluation variables were persisted into a database along with the layout, layout generator evaluations and participant data (Fig. 22).

3.4 Technological Implementation

The structure of a layout generator system may vary depending on the technological approach. The prototype system LaG, which is built for the sole purpose of this thesis, has several components that are typical for layout automation systems. The core system consists of an evaluator with pluggable attribute evaluation managers which has the purpose of quantifying the notion of the attribute on which it evaluates. The evaluator response result is a quality measure of the complete layout as a function based on the results of the specific attribute evaluation managers. This is a typical paradigm for operationalizing layout evaluations. [4] The evaluators have to be implemented with methods that operationalize the function metrics measuring the attribute it evaluates. The LaG system is currently implemented with three evaluation managers; balance, alignment and equilibrium. A further implementation would have been to extend the LaG system with evaluators for other technically defined attributes as symmetry, homogeneity, regularity etc.

The generator in the core system manages the optimization of the layout based on the evaluations. Several optimization approaches have been implemented in earlier research and different algorithms have been recommended. We have used a simulated annealing algorithm approach for the implementation of LaG inspired by Johari et al and Fogarty and Hudson.[29] [31] In research literature the use of genetic algorithm is discussed as a more qualitatively competent algorithm alternative for layout automation [10] [3] as it produces a better optimization for these kind of problems. On the other hand the simulated annealing algorithm tends to be a bit more efficient implemented in the same layout automation environment [10].The LaG system is implemented with a layout generator based on a simulated annealing handling the layout evaluation optimization.

3.4.1 Proof of Concept

Both research question 1 and 2 are questions related to the technical nature of a layout generator. This means that in some way a practical implementation of a layout generator is needed as a basic tool to examine these type of issues further. Therefore we had two alternatives; using an existing implementation of such a system or building one for the sake of the given project. As there are no publicly available complete systems meeting our demands already implemented, we had to build a system from components publicly available and prototype components based on descriptions from related literature.

The LaG system was planned and designed from scratch, based on principles inspired from existing research projects. The metric used in the implemented attribute evaluation managers were inspired by the work of Harrington et al, Bauerly and Liu, Ngo and Ahmad.[11] [6] [5] [10]

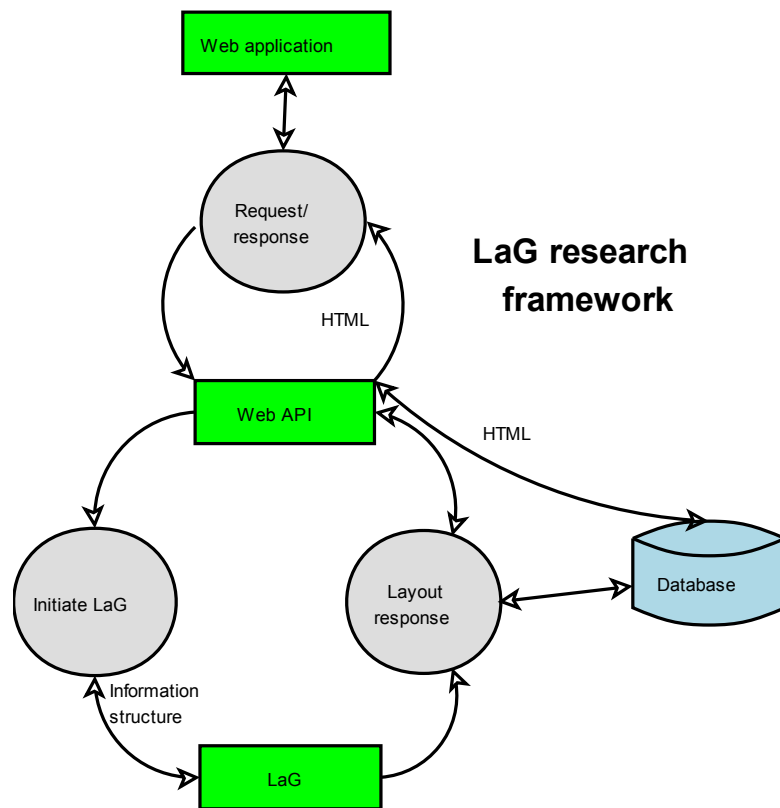


Figure 17: LaG research framework layout automation system

For the layout generator layout optimization inspiration was sought from earlier research that were based on an implementation of the simulated annealing algorithm. A Java implementation of the Cassowary constraint-solver [28] was publicly available on the web and this was chosen for the constraint-solver implementation.

The LaG-system was constructed so that it would be utilizable for different kinds of application. For the purpose of this project we implemented a web API. The web API both serves as a proof of concept of the principles behind layout automation in a web environment and as a test platform for the sake of layout evaluation in this project. One can say that the LaG system is pluggable to any type of Java application that might need the power and application of a layout generator. A further specification of the LaG system will follow in section 3.4.2.

The system implementation was used as a testing platform in this project. We were dependent on gathering data on the layout quality of the layouts a layout generator produced. The main goal was to find out how an extended attribute set in the automated evaluation of the layout in the layout generator affected the perceived quality of the layout generated. We set up two versions of the system for this purpose. One system that generated the layouts real-time and persisted them to a MySQL-database. And one versions that presented persisted layouts to the

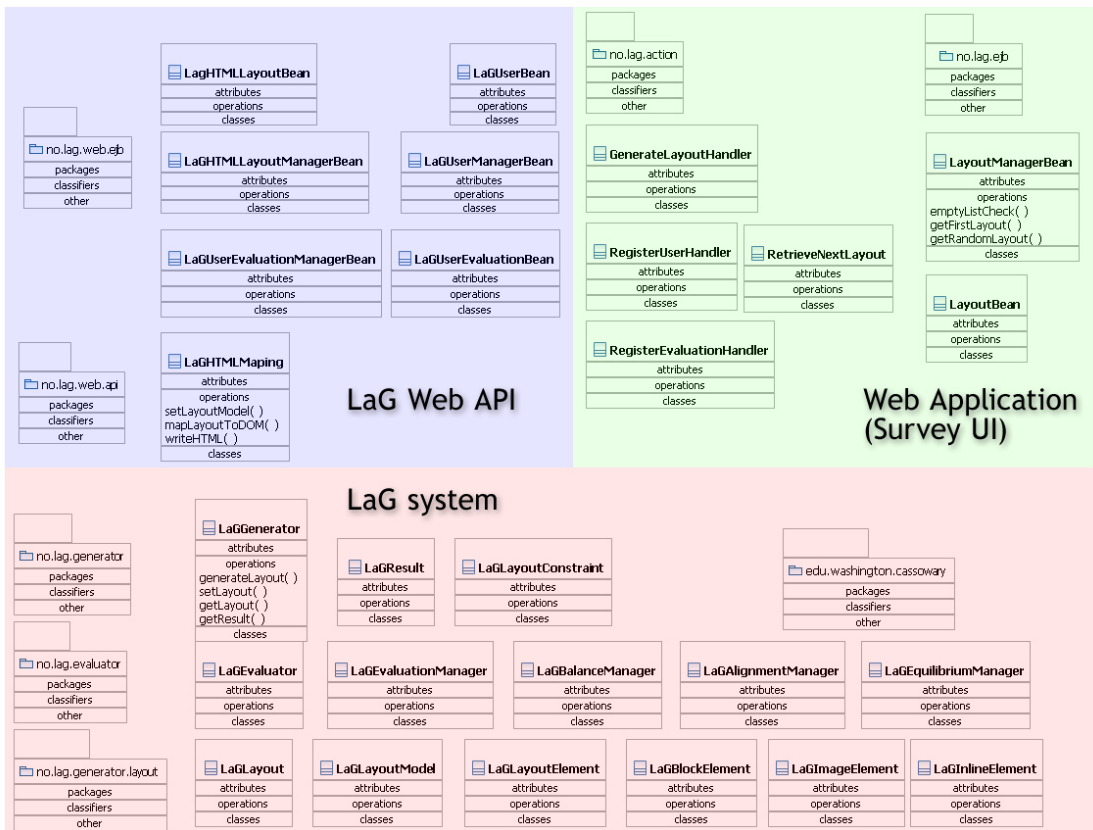


Figure 18: LaG, system research framework object model

viewer in a random order, supplied by an evaluation user interface widget. A modified version of the first system was run to generate 25 layouts for the survey (Originally it only generated one layout and presented it in the web browser). The second version of the system was used in the survey: 25 layouts were presented in a random order and the participant continuously evaluated the presented layout. As the participant had evaluated a layout, a new layout was presented.

As stated earlier the prototype layout automation system LaG is modeled based on the modeling framework presented in chapter 4. The LaG system is used as a research framework in the participant evaluation study and thereby provide a real-life use case of a system model based on the proposed system modeling framework.

3.4.2 Document Layout Generator

Research related to layout automation have resulted in development of several layout automation testing tools. For document layout automation these testing tools range from software features implementing narrow automation of certain aspects of layout to extensive frameworks and systems handling general layout automation. The systems presented in related literature provide solutions for different pragmatic layout problems such as facilities layout planning, VLSI, interface layout and document layout.

The majority of concrete layout automation aid systems are computer-aided design (CAD) type of implementations. [10] Systems related to document layout automation available for research purposes are rare although many practical implementations have been presented in research articles the past decades.

Ahmad presents the general layout optimization IDEAL system which supports the use of subjective and uncertain design preferences. Ahmad approaches layout design related problems in general and the system serves as a testing platform for general purpose research related to layout. [10] Jacobs et al presents a layout template authoring system in addition to a prototype layout engine for the purpose of template-based document layout automation. The template authoring tool is actually a practical a WYSIWYG constraint-editor on which the user defines geometrical areas (constraints) for distribution of certain content. [23] Johari et al presents an early implementation of a similar system related to optimizing pagination in phone book advertising layout called NYNEX. [29] Gajos and Weld have utilized layout automation in the perspective of medium display format layout adaption with their SUPPLE system. SUPPLE is a practical approach to improve the rendition of layout so that it meets the display device constraints. [34] Fogarty and Hudson presents a practical implementation related to the same type of problem. The GADGET system was implemented to provide an experimental toolkit for optimization of layout and interface generation. [31]

Several approaches and solutions have been presented related to web document layout automation. Web document technologies have been widely explored for the purpose of automating the web page layout. Online newspaper layout is a typical use case for such automation tools. González and Merelo presents two algorithmical approaches to optimization of online newspaper layout which resulted in a client-side implementation of a web document layout generator. [16] Gonzalez et al further explored this area resulting in a more efficient client-side implementation of a similar system. [18] The Krakatoa chronicle is another example of an interactive and personalized newspaper layout generator online implemented as a client-side java applet. [30] Zhang and Ren presents a layout generator system for the interfaces of web based transactional pages called CATER. The CATER system is described as a framework solution for field to widget assignment; the automatic generation of UI-widget fields coherent to the database fields to which the information should be persisted. [15] Lohman et al suggests a similar framework for web-document interface generation. The framework is developed to ensure run-time system model dynamics; the generation of interfaces adapts to the system model (Database modeling etc.). [38] Kroener suggests a solution for viewing environment context-based layout generation called the DesignComposer. Context in this perspective means the physical contextual environment of the client presentation medium. [35] Xie et al also presents a system called DRESS for web page layout adapted to the viewing medium presentation surface [26] (These systems are extensively reviewed in section 2.1.1).

Document Layout Generator - LaG

There are many possible approaches to implementing layout generators for automation purposes. Lok and Feiner divides the technical modeling paradigms of such systems into four categories; simple techniques, constraint satisfaction techniques, learning techniques and evaluation techniques. A complete layout generator system may consist of components belonging to different

categories. It is entirely possible to build a layout generator based solely on constraint satisfaction techniques, although this would more or less be a template-based layout system. The research community have generally embraced techniques related to constraint solving and attribute evaluation. [10] [4] Systems based on heuristics and learning techniques likewise.

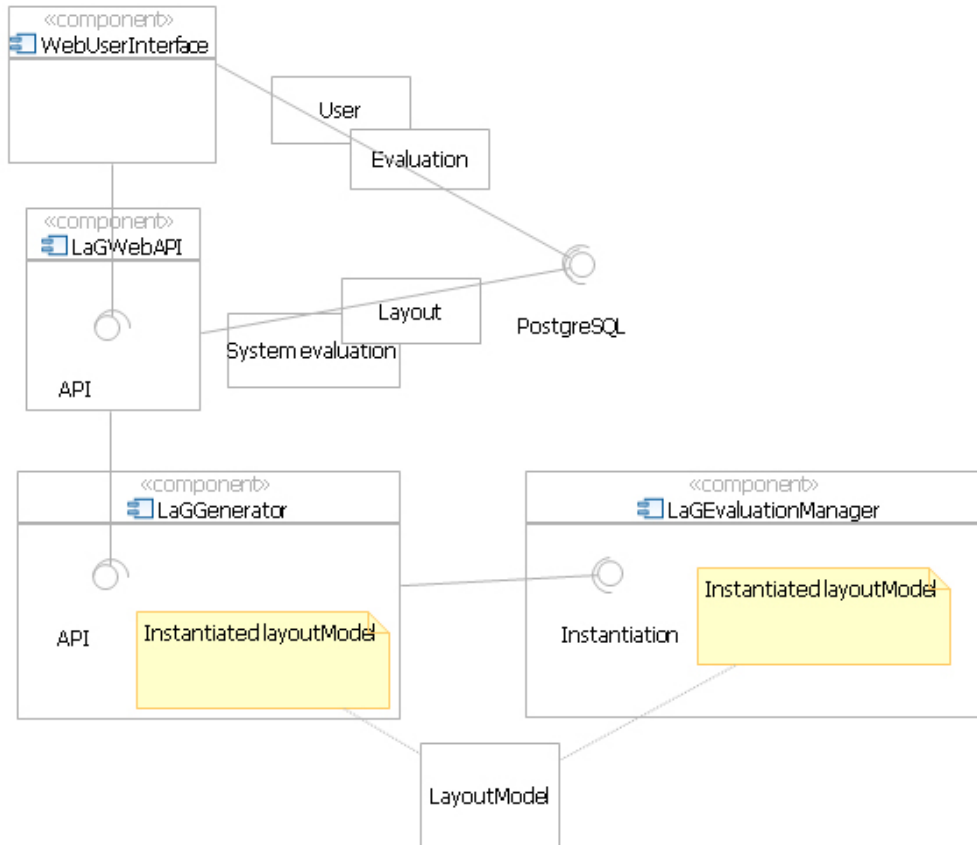


Figure 19: LaG, system research framework object model

In this thesis we propose a general document layout automation system flexible in perspective of preference attribute evaluators and document layout presentation technology; the layout generator system LaG. There are few research oriented layout automation systems for study experimentation publicly available. The few frameworks built for research purposes, like Fogarty and Hudson's GADGET system [31], doesn't meet the specialized requirements of this research project. We needed a layout generator system with support for preference attribute evaluator extensions. Ahmad have implemented a layout generator system with a component called Preference Inference Agent which serves similar purposes. Ahmad's IDEAL system would have been appropriate for general layout research, but the research problem formulated in this thesis include specialized challenges related to document layout. We also wanted to prove that presen-

tation technology independence principles are applicable for such systems, which none of the mentioned research frameworks support. This resulted in a specialized document layout generator implementation used solely for the study related to research question one and two of this particular research project. The implementation of the prototype system LaG were resource demanding, especially when coming to the time resource. Automation systems that aims to relieve the process and workflow from tasks involving the use of human creativity and cognitive intuition are generally complex. Document layout automation systems typically tries to imitate the creative and cognitive skills of the layout designer, (Even though the technologies still are immature) which in itself imply that such systems are generally of high complexity. [10] [4] [24] This is indeed true for systems going beyond the simple techniques described by Lok and Feiner; systems that either are based on learning techniques and AI-related systems [41] or systems based on constraint satisfaction and/or evaluation techniques. [4] [10]

The LaG system is modeled from research literature related to the second two technologies; constraint satisfaction and evaluation techniques. The core layout generator system uses a simulated annealing algorithm which iterates based on overall document layout evaluation results provided by an evaluator system module. The evaluator system model consists of several pluggable attribute evaluation managers internally weighted based on preference (Attribute evaluation weights are set based on attribute preference experiences and suggestions from related literature). The layout administrated by the core system is document-format independent. This means that the layout is merely handled as a model of information structure and presentation layout. The system uses constraint satisfaction for typical ubiquitous document layout requirements as keeping the elements within the presentation surface and no element overlap. These features might as well be handled within a evaluation technique context, instead of using constraint satisfaction.

The LaG system provides an interface to which other systems can communicate. These systems may call the interface and retrieve a generated document layout. For now the document layout generator system only returns a POJO (Plain Old Java Object), but the interface may be extended to map this object to a more appropriate cross-platform and standardized communication technology such as XML, CORBA or specific document formats. [44] [45] [46] We have implemented an external web application that utilize the LaG system as a document layout generator. The web application serves as a proof of concept both for practical real-life expamples and for application used in experimental settings.

Document Layout Generator Core

The core of the LaG system functions as the layout model administrator. An instantiated layout generator constructs the layout model from the information architecture that is to be presented. In the prototype LaG system this information architecture is the same at every instantiation. The same information blocks are provided for the layout generator every time it is run. The information content is irrelevant for the layout generator; it is built to automatically generate a new layout of whatever input it is provided. It is entirely possible to extend the LaG system to support information sources such as XML, database systems etc. to automate the entire data flow from persistence to presentation.

The LaG core is built from the well known object-orientated paradigms of software enginee-

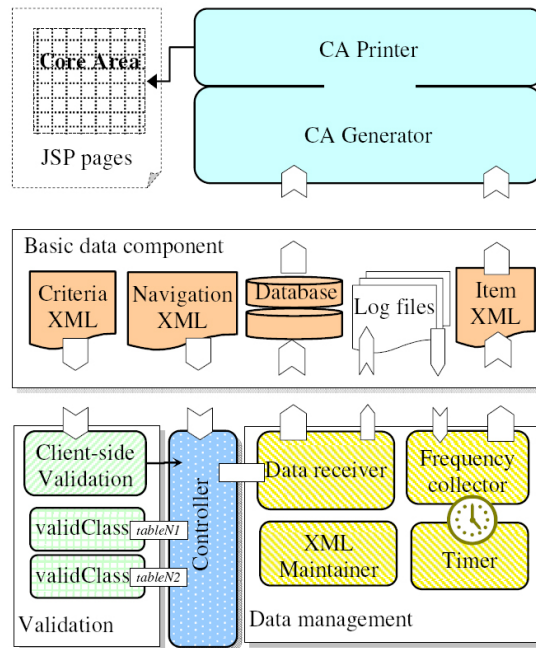


Figure 20: Zhang and Ren, automated layout framework for Web User Interfaces

ring. Object-oriented architecture software design is based on objectification of both functionality and data handling. The LaG core system components such as evaluation manager, layout model and layout generator encapsulates data variables and data manipulation methods in instantiated classes defining these variables and methods. [47] [48] The system architectural design can be described as typical SOA (Software Oriented Architecture), as the layout generator can be set up as a service for applications or other services to utilize. [47] The proof of concept described in this thesis is actually a SOA implementation of a web application that utilize the LaG system as a service.

The service interface of the LaG system is quite primitive. There is no current support for service protocol connection through communication standards as SOAP. [44] [45] [46] The system is set up with possibilities for java RMI (Remote Method Invocation) which enables distributed computing communication through CORBA. Although the support for distributed services is implemented, these capabilities haven't been enabled in the proof of concept web application as it runs on the same application server (JBoss 5.0 application server [49]) as the LaG system. An actual service setup would have complicated the testing phase in experimental settings. As both the application and the server ran on the same application server it is easier to make the web application dependent on the LaG system java application and synchronously deploy both to the application server through the entire testing process. These considerations were made during the implementation phase due to the lack of time resources.

Generally the LaG system core consists of methods and components to administrate initiation, algorithmic iteration procedure, evaluator, layout constraint solving and layout model. Upon

initiation the layout generator core builds the layout model and sets up both evaluator and constraint solver. Further it initiates the actual algorithmical iterations. The layout generator uses a simulated annealing algorithm for maximizing the evaluation scores and thereby optimizing the machine evaluated layout quality. [20]

The algorithmical iterations is the layout automation key process. The layout generator holds an instantiated layout model. During an algorithmical iteration it tries a layout solution close to the current maximized solution (The layout generator holds a layout directly based on the information structure in the first iteration). Before we continue let us revive the technical principles of simulated annealing, the pseudo code: [20]

```
Initial solution s0, energy evaluation e and energy evaluation count k
while k < max(k) and e > max(e) //While stiele iterations &
insufficient evaluation score
  sn := neighbor( s ) //Instantiate neighbor solution
  sn := constraint( sn ) //Constr. solver(Layout generator specific)
  en := evaluate( sn ) //Evaluate
  if en < eb //Higher eval. than the current best; eb
    sb := sn //Copy curr. solution to former best solut.
    eb := en //Update best evaluation
  else
    if P( e, en, temperature( k / max(k) ) > random() //At a probability P
      s := sn //copy current solut. to former best solut.
      e := en
    k := k + 1
return sb
```

In practice it copies the current layout model and tries to slightly move one of the layout elements (A block of information) in the layout model copy. At this moment the layout generator holds two layouts. The new layout is then sent to the constraint solver which manipulates the geometrical variables of the layout elements to meet the defined constraints. Constraint satisfaction, which includes automatic alignment of elements in both horizontal and vertical directions, keep elements within the borders of the parent container and ensuring no-overlap.

The simulated annealing algorithm context is driven by the quantification of a measure parameter connoted as energy. The energy state of every iteration and the current temperature determines whether or not to keep the current solution or not. [20] The main purpose of the simulated annealing model is to allow the algorithm to move further in the search space when reaching a solution that may be a local optimization. By allowing temporarily neighboring solutions of lower evaluation scores, the algorithm may eventually advance in the global perspective. [20] In a layout generator the algorithm uses the quality evaluation as the energy parameter on which it maximizes. The evaluator acquires the layout model from the layout generator core and returns an evaluation score. The evaluation score is metric-dependent; a score calculated from the implemented layout evaluation technique. In LaG the layout evaluator consists of pluggable evaluation managers which handle their respective preference attribute operator; a metric opera-

tionalization of the specific attributes mathematical characteristics. In short terms the evaluation manager quantifies the notion of an attributes perceived presence in a document layout.

The LaG system uses a document layout model to manipulate the information layout. A document layout model serves the purpose of representing the information layout independent of media technology on which to present the information. The model is and should be mappable to any document technology. Upon receiving a response should the applications utilizing the services of the LaG system receive a document format neutral information layout model which the specific application can map to a desired document format. A second possible system service model is to implement the LaG system interface to support diverse standardized document formats or page description languages such as XHTML, PDF or PostScript as request response. This necessitate a LaG system module for handling document format mapping; still it needs a document neutral layout model to work on during the layout generation. The document layout model currently implemented is proprietary for the LaG system. A more appropriate system- and technology-independent model is an important possible extension of the system.

Document Layout Generator Web API

The LaG system communicates with a client application through an API. The web API implemented in the research framework provides methods for requesting a layouts from the LaG system. It provides an information structure for the LaG system to work on and receives a object layout model as response. This object layout model is then mapped to the format which the API supports: HTML and CSS. This means that this specific module works as a communication interface and a mapper from the document format-independent layout model to a standardized document format (Fig. 19).

Document Layout Model

The document layout model enabled in the LaG system is inspired from the object-orientation of common document models and document technologies as DOM (Document Object Model) and PDF (Portable Document Format). At layout generator initiation the layout model is built based on the information structure the LaG system retrieves. The information structure is mapped to a document layout model. The model uses a tree node-structure to represent the information layout. Each node points to instances of layout elements; LaG layout objects which holds the information and methods related to the layout element. For the tree data structure we have used the java-implemented “Data Structure Library in Java” developed at the Center for Geometric Computing, Department of Computer Science, Brown University. The library provides interfaces and classes for programmatic utilization of data structures and algorithms. [50]

The LaG system instantiates each layout element from subclasses of the base class `LaGLayoutElement` which holds general methods and variables that is ubiquitous for any type of layout element. The subclasses `LaGInlineElement` and `LaGBlockElement` inherits the base class in addition to extending with element specific variables and methods. `LaGImageElement` is a subclass of `LaGBlockElement` which holds the raster information of layout image elements. The abstraction structure of the document layout model is inspired from the classification of block and inline elements in XHTML and DOM.[22] Block elements are typically rendered as rectangular layout elements based on the geometrical parameters of the specific block within another block ele-

ment. An entire document can be considered a root block that wraps all the containing layout elements, as the “body”-tag is for the XHTML-document. Inline elements can only contain text or other inline elements. These elements are typically rendered dynamically within the geometrical context (The parent block element) they appear.[22]

These principles are also applicable to typical document standards like PDF. Ignoring PDF interactive elements, a PDF-document contains tokenized elements of PostScript which are handled as directly embedded objects (Directly; a part of the content stream). The embedded objects can roughly be classified as graphic elements and text elements. [51] [52] The service purpose of the LaG system motivates the close resemblance of existing document structures to the structure for the LaG document layout model. This document layout model structure ensures that the service can be utilized by a wide range of document presentation applications (Fig. 19).

Attribute Evaluation Managers

One of the main components of the LaG system is the layout evaluator. The evaluator is instantiated upon layout generator initiation. It administrates the evaluation managers (Attribute specific evaluators) and the evaluations calculated by the managers. In experimental context the evaluator can be set to either use a specific evaluation manager or use the evaluation managers available at run time, a feature that was implemented for the generation of layout samples used in the study. Off course, a distributed layout generated service wouldn't have support for this feature.

The specific evaluation managers administrates the attribute evaluation of the document layout. A balance manager is a subclass of an evaluation manager with an implemented specialized evaluation metric for calculating the balance attribute. All the specialized evaluation managers overrides the evaluation method of the base evaluation manager class; the quantified result of the specific attribute is calculated based on the specific attribute metric. The LaG system currently supports evaluation of the attributes balance, alignment and equilibrium, and the metric implementations of these attributes are inspired from the research of Lok and Feiner, Bauerly and Liu, Harrington et al and Ngo. [9] [6] [11] [5]

The general layout generator model implemented in LaG is based on the layout automation work by Lok and Feiner. They discuss a specialized approach to utilizing a balance operator in layout automation. The concept they present is based on a global visual perception weight map representation of the layout. The WeightMap is actually a gray scale bitmap presentation of the visual weight distribution of the layout surface. Their presumption is that luminant areas have a higher visually perceived weight than areas of low luminance. This luminance distribution is operationalized as a luminance weight map that is utilized for weight distribution calculation. The extent to which the visual weight is evenly distributed on both sides of the horizontal axis determines the vertical balance of the layout and the distribution on both sides of the vertical axis determines the horizontal balance. The sum of both balance operators is a measure for the overall balance in the layout. They have utilized a WeightMap Pyramid processing technique inspired from pyramid representation in image processing to efficiently calculate the balance in both directions. The bitmap representation of the weight map is downsampled based on edge detection.[9] (This is not described in detail, but it appears to be the bandpass pyramid image processing technique) The WeightMap pyramid methodology is actually a sequential bilinear

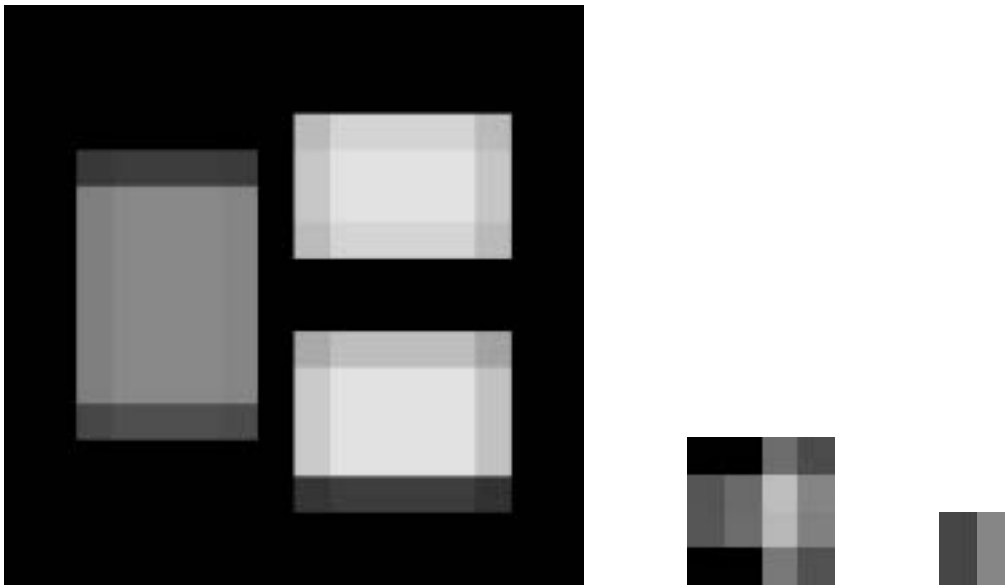


Figure 21: Lok and Feiner WeightMap downsampling for analysis based on pyramid downsampling technique used in image processing

interpolation downsampling of the WeightMap until it consists of an appropriate and real time computationally manageable sample rate. The bitmap presentation used in LaG is downsampled to a 4x4 pixel presentation of the weight map. This makes the pixel matrix suitable for balance quantification as the vertical and horizontal balance can easily be calculated by subtracting the quantified visual weight of one side of the matrix from the adjacent side.

Pseudocode of the balance operator implemented in the LaG-system:

```

init
parameter layoutModel

/***** Recursively iterate the layout model element tree *****/
rootContainer := get root container of layoutModel
globalWeightMap := renderWeightMap ( rootContainer )

/***** Downsample weightMap *****/
array tempWeightMap := globalWeightMap size divided by 2
while tempWeightMap.width >= 4 and tempWeightMap.height >= 4
    tempWeightMap := downsample ( tempWeightMap )

verticalBalanceScore := difference of top and bottom tempWeightMap luminance
horizontalBalanceScore := difference of left and right tempWeightMap luminance
evaluationScore := verticalBalanceScore + horizontalBalanceScore

```

```
return evaluationScore

/***** Recursive weightMap renderer *****/
function renderWeightMap
parameter layoutElement

array weightMap of layoutElement.width and layoutElement.height

if layoutElement is instance of layoutBlockElement
  if layoutElement is instance of layoutImageElement           //If element is image
    image := get image from layoutImageElement
    for every column i < image.width                             //Set background color
      for every row j < image.height                             //of weightMap
        weightMap[i][j] := invert ( luminance ( image[i][j].color ) )
    else
      for every column i < layoutElement.width                 //Set background color
        for every row j < layoutElement.height                 //of weightMap
          weightMap[i][j] := invert ( luminance ( layoutElement.backgroundColor ) )

if layoutElement has child elements                             //Only block elements can
                                                                //contain other elements

  for every childElement
    childWeightMap := renderWeightMap( child element )         //Render childWeightMap
    render childWeightMap from weightMap[childElement.x][childElement.y]//upon weightMap

else if layoutElement is instance of layoutInlineElement
  for every column i < layoutElement.width                     //Set background color
    for every row j < layoutElement.height                     //of weightMap
      weightMap[i][j] := invert ( luminance ( layoutElement.backgroundColor ) )

return weightMap
```

The alignment manager evaluation method is implemented based on the alignment metrics presented by Harrington et al. A measure of alignment may be calculating the geometric distance between natural alignment points: Points of edges of two or more elements it is natural to align in the vertical or horizontal direction. Harrington et al proposes that this is calculated by using a histogram of edge positions (Or center positions). The measure is calculated by sum of distances between neighboring entries in both axis directions. The metric results in high score when the entries are close together; $e = A/(A + z)$ where A is a constant controlling the effect of higher separation distances of neighboring elements and z is the calculated distance. When z is 0 (No

separation) the evaluation is 1, which is top score. The final alignment evaluation is calculated by the normalized sum of neighboring entries evaluation. [11] Pseudocode for the alignment operator:

```

init
parameter layoutModel

integer evaluationScore

recursively iterate the layoutModel containers           //Only block elements can contain
  for every container                                   //other layout elements
    elementList := get container child elements
    evaluationScore += alignmentScore( elementList )

return evaluationScore

/***** Alignment score of elements contained by the samed layoutElemnt *****/
function alignmentScore
parameter elementList

integer alignmentScore
integer penalty
integer alignmentDistance                               //Distance between elements
                                                       //when aligning is req.

while elementList not empty
  layoutElement := pop first element of elementList stack //Pop from the stack
  for every element in element list
    if layoutElement vertical distance to element is less than alignmentDistance
      alignmentScore += ( penalty * normalizedDistance ( vertical distance ) )
    if layoutElement horizontal distance to element is less than alignmentDistance
      alignmentScore += ( penalty * normalizedDistance ( horizontal distance ) )

return evaluationScore

```

The equilibrium operator is based on the equilibrium description of Ngo. The center of mass is calculated as the center of the polar distribution of layout elements based on visual weight and geometric distance. [5] [53] A single layout element equilibrium score is calculated as a linear function of visual weight and euclidean distance (of center point) from the center of mass. The global equilibrium score is calculated as a normalized sum of layout elements equilibrium score. Pseudocode for the equilibrium operator:

```

init

```

```
parameter layoutModel

recursively iterate the layoutModel containers          //Only block elements can contain
  for every container                                  //other layout elements
    elementList := get container child elements
    evaluationScore += equilibriumScore ( elementList )

return evaluationScore

/***** Equilibrium score of elements contained by the samed layoutElemnt *****/
function equilibriumScore
parameter elementList

for every layoutElement in elementList
  integer penalty := normalizedVisualWeight ( layoutElement ) //Vw calculated
                                                            //based on WeightMap
  equilibriumScore += penalty * normalizedDistanceToGeoCenter ( layoutElement )

return equilibriumScore
```

Constraint Solving for Ubiquitous Document Layout Requirements

The system uses constraint satisfaction for typical ubiquitous document layout requirements, such as keeping the elements within the presentation surface and ensuring no element overlap. These features might be handled within a evaluation technique context by penalizing layouts with e.g. overlapping elements hard, instead of using constraint satisfaction. The use of constraint satisfaction is a flexible and robust approach to handling geometrical layout element relations. [28] [4] Badros et al have made a massive contribution to the programmatic utilization of constraint satisfaction principles. First of all their work resulted in a proof of concept technological implementation of constraints in the standardized style technology CSS for the web. Their constraint approach to style sheets was an extension to CSS 2.0, which they called CCSS (Constrain Cascading Style Sheets); a very interesting approach in a practical perspective. Badros et al's research have also resulted in programming libraries for constraint solving in software engineering. The programming library is called Cassowary and provides classes and methods for handling constraints in a programming perspective. The Cassowary constraint solving library have been made available in the programming languages Java, C++ and Smalltalk in the main distribution and has recently been ported to Python and .Net amongst others. [17] [28] [54] [55]

The Cassowary is utilized in the LaG system for the purpose of handling typical ubiquitous document layout requirements. The term ubiquitous document layout requirements points to element layout requirements that in any perspective will degrade the overall quality of the document layout. No overlapping of layout elements is an example of such an omnipresent quality-related requirement. The requirement for layout elements to keep within document boundaries

(Including margins) is another. In the LaG system the Cassowary handles no-overlap requirements, layout element boundaries and layout element offset related to boundaries and other elements. The following constraints were implemented:

- No-overlap: Layout elements on the same structure level can't overlap.
- Layout element boundary: An element is contained and constrained by the parent element.
- Layout element offset: The distance from one element's borders to another element's borders is larger than a defined offset.

Document Layout Persistence

A document layout automation system that provides real time generation of document layout is dependent on service consistence. A distributed server powering the layout automation system may experience performance issues when facing a high amount of client requests. Due to the complexity and general high hardware-resource demand of document layout automation systems it is important to resolve server performance issues real time. The LaG system is equipped with a database persistence component that handles the document layout persistent storage. For now this persistence component have only been implemented for experimental settings, but a distributed LaG-system may utilize the database storage for relieving the application server which it runs on when facing performance issues. The idea is that the LaG-system retrieves already persisted layouts from the system when the request load is reaching capacity and the information to be presented is similar to an earlier request (A use case might be the presentation of a search result based on two or more similar search queries on information that haven't changed).

The persistence storage has been practically implemented in experimental settings for this thesis (Fig. 22). The layouts that are presented in the study practitioner survey were generated in advance and persisted as XHTML-code to a PostgreSQL database storage. During the survey layout evaluation session the participants were presented layouts directly retrieved from the database. One might argue that the system performs effectively enough to present real time generated layouts during the survey, but this is impractical related to the study design of the experiments. The study data consistence relies on that each particular document layout is evaluated by several practitioners; this ensures that practitioner's subjective preference is suppressed for the sake of the global practitioner population preference. Real time generated layouts implies that the participants would have evaluated 6 different sets of layout samples.

The utilization of document layout persistence in the web application used in the experimental settings is provided as a proof of concept related to the general utilization of persistence storage in document layout automation systems. A SOA and server-side implementation of a document layout automation can be equipped with advanced features such as persistence storage in a larger extent than client-side implementations, especially for web-technology based layout automation systems. The LaG system is based on a completely different architecture than other document layout automation systems as it can utilize service-related features that extends the basic document layout automation software presented in related research. The client-side web-technology based systems presented in related literature is incapable of utilizing distributed services in the same manner as it is relying on the features and restrictions of the client application which it runs on. [30] [16] [18]

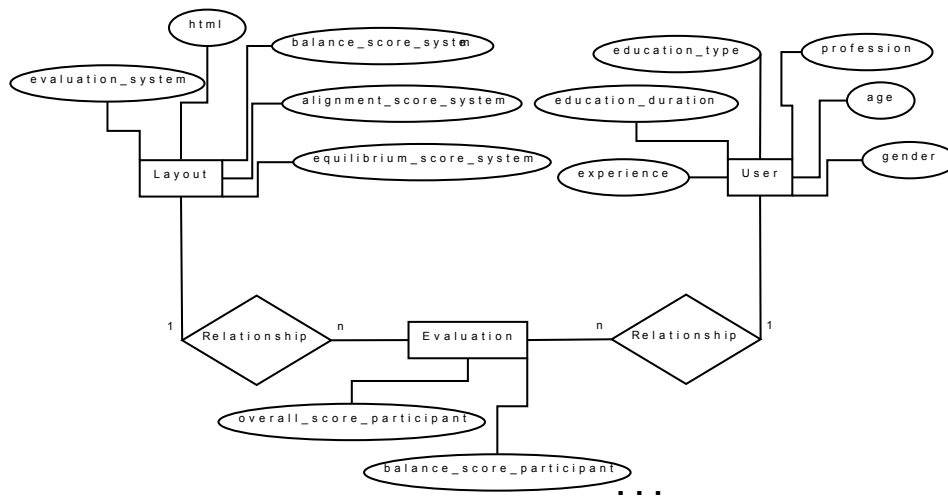


Figure 22: The research framework persistence ER. (The additional attributes of entity evaluation are described in section 3.5.1)

Document Layout Generator Interface

The current LaG interface is merely implemented to serve the purposes of the study. The interface provides methods for...:

- ...initiating the generator.
- ...setting the base layout (The information structure) upon which the generator should generate new layout from.
- ...adding evaluation managers to the evaluator.
- ...getting the evaluator.
- ...getting and setting evaluation results.

The LaG constructor initiates the layout generator, the evaluator and a layout model. A constructor with a provided base layout model (The initiation information structure) as parameter has also been implemented. This is functionality that haven't been used in experimental settings, but clearly a feature that a layout automation service needs to provide to the clients.

Proof of Concept Web Application API

The web application API implemented for the survey is a proof of concept related to the practical application of the LaG system. The survey web application user interface uses a web application

API with which it communicates. The survey web application sends requests to the web application API, which returns a generated XHTML document layout of the use case information. A callback servlet in the web application handles the request and response of the client interface. The web application uses the web application API for ORM (Object to Relational Mapping) for receiving layout and layout evaluation data wrapped in the same object. In the use case for the survey we implemented the layout generator to work on the layout of a product presentation. The products in the product presentations were described by images, description text and price text. Each product block were automatically layed out by the LaG system, mapped to XHTML and CSS by the web application API and presented by the web application user interface.

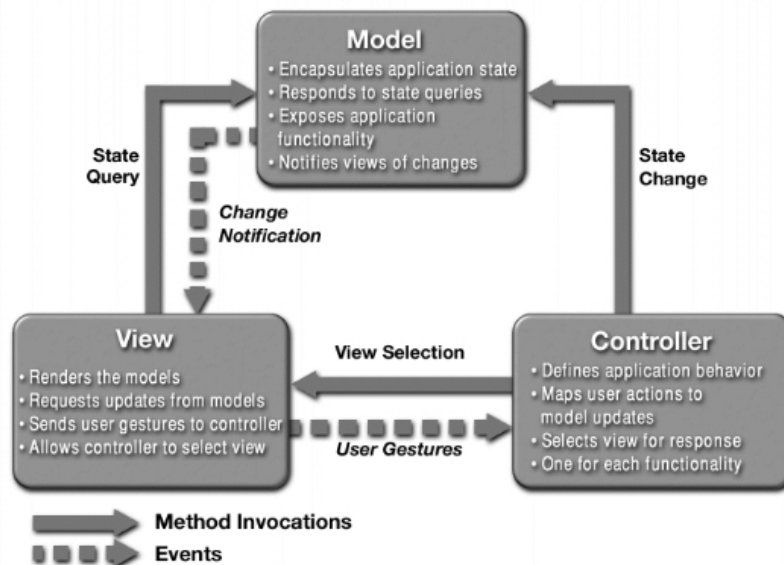


Figure 23: Sun, general java MVC-pattern model [1]

The survey web application is a typical implementation of a layered architecture system model. The web application uses the MVC (Model - View - Controller) paradigm; the user interface communicates with a controller which handles requests and the controller communicates with the business layer (Model) which serves the data (Fig. 23). [47] [48] [56] The web application used for the survey the web interface uses JSP (Java Server Pages) files utilizing the power of a JSP-extended tag library called JSTL. This is further described later in this chapter. The web application is implemented with request type-specific handlers (Java servlets) as controllers. Whenever a participant in the study clicks to finish evaluation of the presented layout, a handler is collecting the request and instantiates a new layout (A LaGHTMLLayout-object) which it serves to the client as response. In the same manner a client request callback is implemented for handling the participant user information registration upon survey initiation. The JSP template for presentation of the response layout ensures that the XHTML-layout is presented and layed out.

The web application is technologically fundamented on the Java EE platform. It runs on the

application server Jboss 5. [49] [48] The view layer is based on JSP and JSTL technology, the controller layer is implemented on java servlet technology and the model layer is based on EJB (Entity Java Beans) technology. The web application API uses Hibernate for persisting document layout data to and from a PostgreSQL database server. [56] [1] [57] [58] [48]

Survey Web Interface

The web survey interface utilizes the server-side web application API. The web survey interface is a typical view-implementation based on JSP and JSTL server-side scripts which handles request responses. The web survey interface implementation utilizes AJAX-technology to minimize the user-interaction disruptancy.

A survey participant starts the evaluation session by registering background variables. The user fills in a form and click the “submit”-button to register the details related to this specific participant. After registering the participant details the participant faces an evaluation user interface. This UI is separated in two parts; a defined area for the user evaluation form which the user fills in for every layout presented and a defined layout presentation area. The background colors clearly separates these two blocks so that there is no doubt which geometrical area to base the evaluation on (This was additionally utterly specified verbally upon survey initiation). The layout presentation block is colored white and the evaluation form block is colored gray so that attention is driven towards the layout presentation area (Fig. 30).

3.4.3 Technical Implementation of Layout Generator in the Survey

JQuery (A client-side script library for interface logic and functionality) handles the client-side server requests and responses in the survey web interface. JQuery is based on AJAX-technology and allows the interface automatically load request responses directly in defined areas of the web interface (E.g. loading the response in a certain HTML block element). [59] Whenever the user clicks to submit an evaluation a client-side javascript uses the JQuery library to send the request. The response is yet another generated XHTML-layout which a JQuery-callback handles and loads into the layout presentation block. The utilization of AJAX in experimental settings such as in this survey reduces the continually experiment disruption by reloading the web-document upon HTTP-requests.

The web application handles the client responses and registers the data provided from the interface. The web application API with which the controllers (handlers) of the web application communicates is implemented with ORM-technology through the Hibernate API. These handlers merely instantiates certain ORM-enabled EJB classes and utilize commit methods in the API which persists the specific object to the Postgre SQL database. [48]

3.5 Data Collection

The study design provides an experimental tool to help answer the research questions. Research question 1 is aimed towards the utilization of attribute operators in document layout automation, how additional preference attribute operators affect the general quality of generated layout. To operationalize such a research problem we needed to quantify the notion of specific preference attributes of layout and the general notion of document layout quality. The notion of quality has to be quantified in two perspectives; the mathematical quantification of quality utilized by

the layout automation system and the quantification of visually perceived quality. The layout automation system measure of quality is a specific component of systems based on evaluation techniques. [4] The measure of visually perceived quality is a specific methodological tool for use in the experimental setting of the study.

The web application API system ensures the persistence of the system quantification of layout attributes and general system measured layout quality. There are three specific document layout attributes that are investigated in the study; balance, alignment and equilibrium. The system quantifies these attributes based on the metrics described earlier in this section. The overall quality is quantified as a function based on a weighting metric inspired from the regression equation presented by Ngo and the work of Harrington et al. [5] [11] The measured attributes and measured overall quality are persisted to a PostgreSQL-database.

The web application API handles the data collected from the study survey. The web interface evaluation panel enables evaluation of the attributes balance, vertical balance, horizontal balance, symmetry, alignment, regularity, equilibrium and efficiency (These have been described and defined earlier in this chapter).

A separate manually filled data sheet has been used for the survey interview. The answers of the questions were archived with a written participant number generated from the web application API when the participant registers the participant background variables.

3.5.1 Data collected

The data collected is separated in three parts: data collected from the web application API when generating the 25 layout samples used in the survey, data collected from the participant (Background variables) upon survey initiation and data collected from the participant's evaluation of one specific document layout.

The data collected from the web application API when generating layout samples were:

- The layout id (Sequence auto generated id by the DBMS).
- The document layout mapped to XHTML-code.
- The system overall evaluation of the specific layout.
- The system balance evaluation of the specific layout.
- The system alignment evaluation of the specific layout.
- The system equilibrium evaluation of the specific layout.

The data collected from the participant upon survey initiation (Background variables):

- The participant id (Sequence auto generated id by the DBMS).
- The participant's profession.
- The participant's experience in the design industry.
- The participant's education.

- The participant's education duration.
- The participant's age.
- The participant's gender.

The data collected from the participant's evaluation of a specific layout (Attribute scores and overall scores evaluated on a scale from 1 to 10):

- The evaluation id (Sequence auto generated id by the DBMS)
- The evaluation of balance.
- The evaluation of vertical balance.
- The evaluation of horizontal balance.
- The evaluation of symmetry.
- The evaluation of alignment.
- The evaluation of regularity.
- The evaluation of equilibrium.
- The evaluation of efficiency.
- The evaluation of overall quality.

The data sheet and interview questions of the participant interviews are provided in appendix D.

3.5.2 Collecting Data

Data used in the study were collected both manually and automatically by the experimental support systems such as the web application API. The system layout evaluation data were collected automatically by the web application API. The web application API provide methods to commit the data collected from the system evaluation. The software utilizing the commit method of the web application API only instantiates a LaGHTMLLayoutBean (EJB class) and sends this as a parameter through the commit interface for this specific class. This commit method may be called from any software implementing the web application API (E.g. a servlet handling requests from the web application used in the survey) [48] [58]

Data collected from the participant registration form were also collected in the same manner as the layout evaluation. A servlet handles the response of the web application by registering the participant data in an instantiated LaGUserBean (EJB class). The instantiated object is committed to the database by using the class-specific commit-method of the web application API.

Data collected from the participant evaluation of a specific layout were also collected in the same manner. A server side handler (servlet) instantiates a LaGEvaluationBean with the data provided in the request which is committed to the database through the web application API.

The data collected from the participant interviews were manually written down on the data sheet used during the interview.

3.5.3 Persisting Data

The data collected needed to be digitally persisted for further analysis later in the study. A PostgreSQL RDBMS was used for data storage. The relational database and RDBMS provided a flexible platform for data mapping to the SPSS data analysis tool used. The web application API used Hibernate and ORM for data persistence. Object to relational mapping tools such as Hibernate makes the system to database communication much more flexible and robust. The Hibernate API administrates the data transaction independent from the data access layer. Hibernate secures data access layer portability without changing the programmatic business logic of the system. [48] ORM is set up and described in Hibernate mapping configuration files independent from the rest of the system. The web application API provides methods for committing through Hibernate based on ORM. These methods supports committing the classes used in the survey web application and are potentially extensible with any type of object persistence as long as the ORM is configured. [48]

The persistence of data collected from the participant interview was technologically limited to the manual transcription of the statements on a plain old piece of paper.

3.5.4 Handling Data for Analysis

We used SPSS as the statistical analysis tool for this study. SPSS supports a wide variety of data formats including Excel, SQL, CSV etc. We implemented a script to map the data collected from the RDBMS to the CSV file format. Each line of the CSV-file were data identified by the evaluation id as both the participant data and the layout data is functionally dependent (Based on database theory) on the evaluation id. (And clearly the evaluation data is functionally dependent on the evaluation id) The generated CSV-file was imported in SPSS and the collected data were ready for statistical analysis.

4 Layout automation modeling framework

4.1 Layout automation system modeling

The research literature presents many different approaches to layout automation system modeling. Few research articles have discussed system model generalization when using evaluation techniques in layout automation systems. A thorough examination of the research literature proposed systems, prototype systems, modules and components related to evaluation techniques in a layout automation perspective quickly reveals several architectural similarities. Research is evolving around the same modeling paradigms, but it seems as though each research article proposes a proprietary set of system modules and components for programmatic implementation of the same functionality. There are several examples of layout automation system model proposals, but each of the systems aims to solve a particular problem or a set of problems of the same problem domain.

We propose a layout automation system modeling framework for the modeling application of a wide range of layout automation-related problems. The modeling framework is generalizable for layout automation system modeling of layout automation research frameworks, distributed layout automation systems, layout automation software etc. Based on the experiences of setting up a layout automation system and reviewal of several research proposals presenting such systems we have derived a general modeling framework that can be utilized as basis for specific layout automation system modeling. The framework is built upon abstracting knowledge from research literature, organizing this knowledge and deriving the common architectural paradigms arising from this derivative analysis.

4.2 Modeling framework

The notion of the term framework is quite vague; in research a conceptual framework is the presentation of a preferred approach to an idea, in software engineering a framework is the re-usability of architectural and programmatic design. The American Heritage Dictionary of the English Language (Distributed through dictionary.com) defines the term framework (In an academic context) as *a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality*. [60] The layout automation system modeling framework presented in this thesis consists of the systemization of typical modeling paradigms detected in a wide range of research literature related to the area of layout automation. The research framework represents a set of possible component modeling paradigms, description of general components and a generalizable system architectural pattern; these paradigms don't in any way constrain the general modeling of layout automation systems, but rather enables a more efficient development phase as typical modeling structures are already established and formalized.

The system modeling framework is in particular based on the work by Lok and Feiner [4], Fogarty and Hudson [31], Gajos and Weld [34], Zhang and Ren [15], Ahmad [10] and Kroener [35]. Other research projects that have presented related conceptual knowledge applicable in such a modeling framework are Bateman et al [24], Kamba et al [30], Jacobs et al [23] and Badros et al. [17] [28] The knowledge derived from these articles are further described in the following sections.

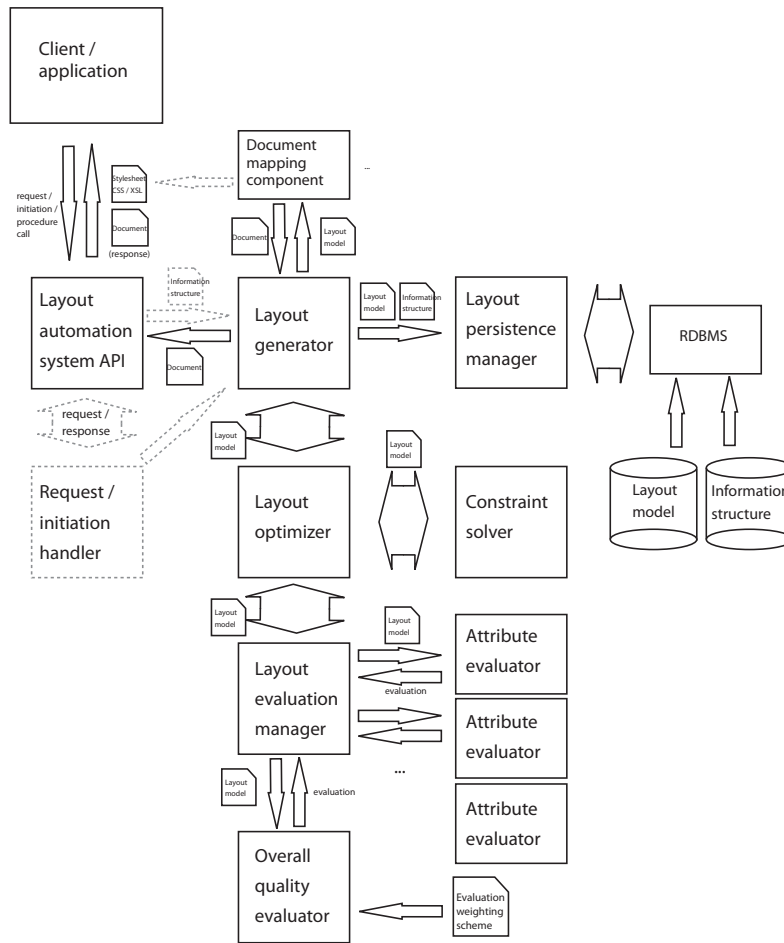


Figure 24: Document layout automation system model framework

4.3 General document layout system model framework

The framework describes two aspects of architectural design; the data administrated by a layout automation system and the general functionalities of a layout automation system. [47] Typically a layout automation system consists of modules or components administrating and specializing in particular tasks related to layout automation systems. The research of Ahmad, Zhang and Ren, Kroener and Kamba et al all thoroughly describes system models applicable to certain aspects of layout automation. Ahmad presents the IDEAL system based on a system model for a layout automation research framework. [10] Zhang and Ren presents the CATER system model for handling the automatic layout of transactional web pages [15] and Kroener and Kamba et al present novel systems for web document layout automation. [35] [30] Other research projects proposes systems and components applicable for certain tasks related to layout automation. Fogarty and Hudson presents the GADGET framework; an optimization toolkit utilizable in a layout automation context [31], Lok et al presents an evaluation abstraction methodology for

<i>Interface technologies</i>	<i>Description</i>	<i>Communication protocol</i>
SOAP / XML-RPC	XML-based object mapping; suitable for XML-based format interchange	HTTP
CORBA (ICE)	Distributed object mapping between different system implementation technologies (E.g. Java RMI)	GIOP (General InterORB Protocol)
DCOM	Proprietary Microsoft technology for object-oriented communication	DCOM remote protocol
General API's	Proprietary local API's or distributed API's through HTTP etc.	Local application object binding through programming API libraries or HTTP

Table 2: Layout automation system interface technologies

attribute evaluation managers through their BalanceManager implementation [9], Jacobs et al demonstrates how to utilize templates in layout automation [23], Badros et al both provide constraint satisfaction framework utilizable by layout automation systems [28] etc. (The systems and components are extensively described in chapter 2)

4.3.1 Layout generator

The layout generator is the core component of a layout automation system. The generator can generally utilize a range of components based on any of the four layout automation techniques described by Lok and Feiner. [4] A layout generator utilizing evaluation techniques are dependent on a few core components; a constraint solver, a layout optimizer and a layout evaluation manager. These components among the others illustrated in figure 24 will be extensively described in the following sections.

The layout generator administrates every aspect of the internal layout automation from a client initiating the generator and passing information parameters to make a presentation of, to the layout of this information presentation is generated and responded back to the client initiating the system. A request may be a server-side request by a client application, a layout automaion application interface call etc (Table. 2).

Several communication technologies can be utilized in such a system. Systems can be grouped in two categories; distributed systems and services, and local applications and software. The first group utilizes protocol-based technologies enabling network communication and the latter utilizes application to application communication or imported local application API libraries.

The layout generator administrates the evaluation optimization of a layout. This evaluation optimization is dependent on the quantified evaluations of layout characteristics, thus it needs

a layout evaluation manager administrating the evaluations. If the layout generator utilize the strength of constraint satisfaction technologies, a component for handling constraints is also needed. The layout generator general procedure is to build an initial layout based on the initial information structure. This layout is passed to the optimizer which optimizes the layout by using the layout evaluator. If the system is using a document format mapping component, the layout model is mapped to the specific document format. A prototype layout generator procedure description (Pseudocode):

```

init \\Initiation
  parameter informationStructure \\Parameters
  parameter documentFormat
  layoutModel := mapToLayoutModel ( informationStructure ) \\Map information to model
  layoutModel := optimizeLayout ( layoutModel ) \\Optimize the layout evaluation
  if persistence is enable
    persistLayout ( layoutModel ) \\Persist to db or file
  document := mapToDocument ( layoutModel, documentFormat ) \\Map from model to format scheme
  return document      \\Return document

```

4.3.2 Layout optimizer

The layout optimizer uses a general optimization algorithm which minimizes or maximizes the evaluations made by the layout evaluator. The component can either utilize proprietary technologies or use existing optimization libraries or components such as the GADGET-framework by Fogarty and Hudson. [31] There are also a range of libraries developed for different programming languages and implemented with optimization algorithms applicable for layout automation publicly available. E.g. we used a publicly available java-implemented simulated annealing optimization algorithm (No name available, pseudocode described ref. 3.4.2).

4.3.3 Layout evaluation manager

The layout evaluation manager is one of the most important components of a layout automation system. It handles the administration of layout characteristics evaluators (Preference attribute evaluators) and the general metric and calculation of the overall quality of a layout. The evaluation manager harvest evaluations of certain characteristics from the attribute evaluators and calculates the overall quality based on the overall quality metric implemented. Some metrics for overall quality measures have been proposed in research articles; Harrington et al presents the most robust and promising overall calculation metric (Eq. 2.24) [11], while other research articles presents straight forward sum of evaluations metric. The latter are rarely described in the research articles, but can be mathematically presented as a function of the evaluations in the following manner; $O = \sum w_i e_i$, where O denotes the overall evaluation, w the weight and e the attribute evaluation.

Attribute evaluators

An attribute evaluator is the component evaluating one or more specific characteristics of a layout. The attribute evaluator is implemented with a metric or procedure for quantifying the notion of the specific layout characteristic. We have already discussed a number of proposals for

operationalizing the layout characteristics; the BalanceManager of Lok et al [9], the operationalizing metrics of Ahmad [10], Harrington et al [11] and Faria and de Oliveira [8] etc. E.g. the LaG system has been implemented with attribute operator components for the evaluation of balance, equilibrium and alignment.

Overall evaluation handler

The overall quality measure is calculated by the overall evaluation handler. The evaluator uses the attribute evaluators to quantify specific layout characteristics and passes these evaluations on to the overall evaluation handler. The overall evaluation is a function of the attribute evaluations; an implemented metric calculates the overall quality evaluation based on the attribute evaluations. An example of an overall evaluation metric is proposed by Harrington et al in equation 2.24. [11]

4.3.4 Layout persistence manager

Some of the research articles presents components for persisting generated layouts and harvested information related to the generated layouts (E.g. user interaction tracing[30], layout scoring(The LaG system) and expert layout evaluations [10]). This feature is applicable for the reuse of layouts. A layout of the same or similar information structure may be reused by the system to save computation time. An example is the presentation of a search for books on Ebay based on the same (and widely used) long tail search query; instead of generating a new layout an existing layout is used as long as the information structure to be presented haven't changed dramatically. (E.g. the search query gives a completely different result as many new books are available)

4.3.5 Constraint solver

The constraint solver handles the constraint satisfaction in the layout model. E.g. typical spatial constraints as keeping layout elements from overlapping, keeping layout elements within the document boundaries etc and abstract constraints as keeping the label of an element close to the element itself. Badros et al has proposed a flexible constraint solving toolkit called the Cassowary for the purposes of spatial constraint satisfaction. [28] The programming library has been implemented in a wide range of programming languages and utilized in several research projects (Such as the BalanceManager of Lok et al [9] and the LaG system presented in this thesis).

4.3.6 Layout model to document mapping

The layout model to document mapping component maps the layout model to the document format requested. The interface API restricts the document format based on which format mapping components that are available in the system.

4.3.7 Layout automation system API

The system communication interface handles the communication with the client. The layout automation system API should provide methods for initiating the layout generation based on the information structure and document format passed as API parameters. The technology to base such an interface is dependent on the system scope; a service oriented system scope needs a distributed interface for clients to communicate through, an application system scope might only

need a programming API to communicate through (Communication technologies described in table 2).

4.4 Data

4.4.1 Information structure

Upon initiation the layout automation system needs to receive information to lay out. This information has to be organized in either a semantic structure or an abstracted inter-element relation structure. (Much in the same manner that abstract constraints work) The system can't detect these structures automatically just by analyzing the stream of media elements it receives upon initiation. This is an entirely separate research field; the derivation, abstraction and analysis of connotation and semantics from media elements such as images, video and text.

This means that the information stream passed to the layout automation system needs to be logical structured in a machine interpretable format. This further implies that a structured and standardized scheme to present information and the information element's abstract relations would have been of great use. Lok and Feiner discuss the utilization of abstract constraints and describe an example of tokenizing and expressing these constraints. [4] As the lingo for expressing and the scheme tokenizing the constraints and elements is formalized the system is competent to automate the layout of the information. There is also a need for formalized scheme for expressing layout elements and their properties. E.g. a title may be annotated *title* and an image may be annotated *image*. An example of annotation (tagging) of layout elements is the XHTML-standard and the DOM-representation of such documents. The DOM-representation is actually an object-oriented mapping of the XHTML-document content and environment. [22]

Zhang and Ren actually describes a similar mapping and standardization of information. The CATER system enables mapping from database schemes to a proprietary XML-scheme. The CATER system generates and lay out UI-widgets (HTML user interface form and input-elements) for the web based on the XML-representation of the database scheme. [15]

Fu and Liu proposes a gateway from HTML to XML which is interesting in this perspective. First of all they present an interesting approach to semantically annotate XML-elements mapped from the HTML-document. They propose interesting solutions for detecting layout blocks in HTML documents, semantical annotation of HTML content and technologies applicable for mapping of XML-based information, all of which is applicable for a layout automation system. By detecting blocks of content the system presented is actually structuring and analyzing the layout of the HTML-document; a great feature for a layout automation system. They also describe a semantical annotation of XML-documents based on the HTML-content. As mentioned earlier, the semantical structuring of the information is a useful feature for a layout automation system.

The last and maybe the most useful feature presented is the mapping technology used; XSL (Fig. 25). [61] XSL, XSLT and XSL-FO are standardized and widely adopted technologies for styling and transformation of XML-documents. [22] The utilization of XML for information presentation and XSLT for transforming from the information presentation format to a layout model is straight forward and robust (XSL or XSLT processor is available for most programming languages).

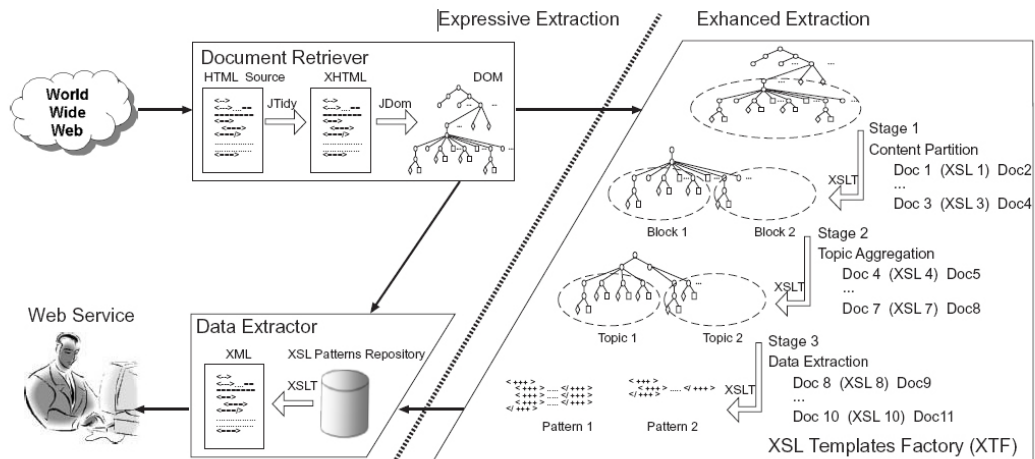


Figure 25: Fu and Liu, a gateway from HTML to XML

4.4.2 Layout structure

For the internal processing of the layout generation a mapping from the XML-representation of the layout model to a object-oriented representation is reasonable; either a DOM-representation, which is generally resource demanding, or a proprietary representation (As utilized in LaG; JDSL [50]).

4.4.3 Output document layout

Enabling a second transformation from layout model to document format after layout generation further ensures a standardized formatting technology throughout the workflow of the layout automation system. By implementing XSL and XSL-FO-processors in the layout automation system it is possible to map the layout model to a wide range of document formats such as XHTML (Or any other XML-based document formats), PDF, PS or even RTF. Specialized mapping tools can also be implemented to enable mapping of the layout model to other document formats.

4.4.4 Evaluation weighting scheme

The evaluation weighting scheme is merely an attribute evaluation weighting configuration. This can be provided to the system either through a file containing the settings (XML or any other typical settings-file format) or by a database persisting the settings. The file or database scheme defines the attribute evaluators to utilize and their weighting. The overall evaluator uses these weighting numbers to calculate the overall quality measure based on the attribute evaluations. (It is entirely possible to use a hard-coded weighting scheme, but this is not recommended as the system should be easily configurable)

4.5 Extended modeling

Some aspects document layout automation not described by the modeling framework of this thesis might be reasonable to implement depending on the system purpose. There are several other functionality components that are utilizable in a layout automation perspective, but not

essential to the modeling paradigm of the modeling framework.

4.5.1 Context analysis

Layout generation based on context analysis is extensively described in several research articles. Context-based layout automation related to the analysis of the environment variables affecting the presentation. [35] These environment variables are related to the presentation environment (Display / presentation surface, technological limitations etc.), user / user group environment (User type), information environment (Information semantics) etc. Layout automation based on context analysis aims to adapt the presentation layout to certain environmental variables.

A layout automation system handling context analysis is dependent on a context retrieval component retrieving the context information from the client using the system. Kroener has proposed how to utilize such a component in a layout automation system (Fig. 11). Kroener addresses the utilization of presentation environment information for context analysis, as the vast majority of context-analysis research related to layout automation does. [35] Gajos and Weld presents the SUPPLE-system; a UI layout generator utilizing the user information and presentation environment information to generate user interfaces by user interaction analysis. This system is extensively prototyped and a video presentation is available online: [33]. [34] Other researchers presents related aspects of context analysis and modeling; Schlunbaum presents a user-model based UI-generation [62] [4], Ishak and Feiner presents content-aware (Information semantic analysis) layout for automatically arranging desktop windows [27] etc.

4.5.2 Expert evaluation

Ahmad presents an expert evaluation component for the use in research framework layout automation systems (Fig. 26). Ahmad proposes the use of a preference discovery agent to analyze the expert preferences related to certain information layouts. The knowledge acquired from the expert evaluations are persisted for use by a preference inference agent, a component quantifying the preference attribute weights. [10]

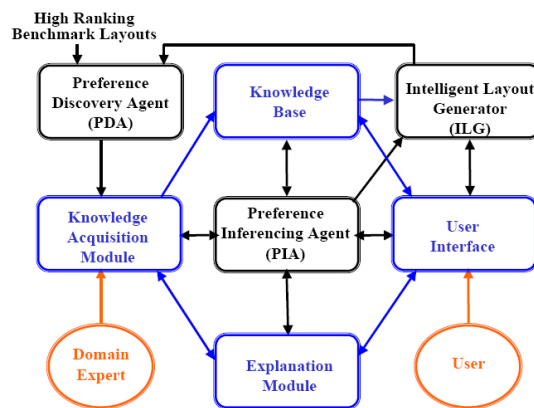


Figure 26: Ahmad, IDEAL system model

Ngo has also provided research related to how the expert preference evaluation can be quan-

tified and analyzed to generate regression equation of the expert acquired evaluations.[5] The research is actually very interesting in the scope of layout automation system expert evaluation operationalization, although it doesn't provide any actual practical modeling or prototyping of the utilization of expert evaluation information.

5 Results

5.1 Practitioner Interview

5.1.1 Efficiating work tasks in the process of composing layout

There may be several tasks and processes in the workflow of producing layout that are suitable for efficacy increase. The practitioners were supposed to consider whether or not there was room for increasing efficacy of these processes based on statements presented related to specific work tasks.

There was a large consensus related to the fact that their field of work consists a high degree of task plurality. The practitioners state that the projects they are involved in are characterized by diversity when coming to the use of creative skills related to layout. They state that they are often working with regulated layouts; either regulated in the notion of geometrical inflexible layouts or regulated by the standards and demands of the customer client. An example of a geometrical inflexible layout is a layout consisting of a large number of semantically similar graphical elements laid out in a repetitive manner. They also state that they work with many design projects characterized by creative freedom and a high degree of autonomy.

The participants generally agreed upon that they regularly work with layouts with graphical elements laid out in a repetitive manner. The general response was “Yes, I often work with templates and grid-based designs and repetitive tasks related to the layout of the graphical elements”. One of the participants stated “Absolutely, more or less all the time...” although he / she also stated that he / she worked with many creative tasks as well. Examples of such tasks were advertisements (Often because of requirements in a style guide or required consistency in the visual communication), print, brochures, web articles, web menus etc.

The participants also generally agreed upon that they worked with tasks characterized by a high degree of creativity. Generally the participants state “Some work tasks are creative, it usually depends on the client”. One of the participants also adds that the room for creative freedom is more present in the design aspects as styling, color composition rather than layout. Another participant states that the initial phase is often more characterized by creativity, e.g. sketching ideas and tweaking to find a communicative edge.

We asked the participants to consider which type of work tasks (Provided as examples) were suitable for automation (For higher process efficiency). The participants more or less agreed upon certain work tasks that were more suitable for automation; e.g. general grid-based design might be suitable for automation because of the rule-based manner of laying out elements. A grid is generally a set of rules (More specific constraints) on which to base the layout on, which makes this assumption quite reasonable. Further the participants agree that use cases describing the layout of articles in an online newspaper, list of results of a search for books in a web page and layout of advertisements in a printed publication might be suitable for automation. All these tasks generally consist of similar and repetitive subtasks. The participants generally agree upon what kind of work tasks that might be suitable for efficiating by the use of support automation tools. The contradictory use cases, layout of a book front cover and layout of a rock festival poster, were agreed upon unsuitable for layout automation.

5.1.2 The use of well-established tools in the process of composing layout

In the second part of the interview we presented statements related to the existing utilization of layout automation tools. The participants were asked to consider and elaborate three statements.

The first statement was “I use general tools (Grid-systems, templates etc.) for easing the process of composing layout”. All the participants answered that they regularly use support tools when composing layout of graphical products. These tools are by the participants characterized as convenient because they ease certain design process tasks. Examples of design process tasks as described by the participants; “Ease design changes requested by client..”, “Implementing relatively identical designs adapted to different formats”, “Sketching drafts” etc. The participants generally stated that the Adobe products were the preferred choice of software environment. The Adobe products (E.g. Adobe InDesign, Illustrator, Photoshop etc.) are generally implemented with support tools that helps the designer in the process of layout composition. Examples mentioned by the participants were template support, grids and guides, automatic text-wrapping, pagination etc.

The second statement the participants considered was “I use automation tools for layout composition in my profession”, a statement closely related to the third statement “If you agree to the previous statement; which tools do you use...”. During excogitating this statement the participants were generally surprised by the extent to which they already increase process efficacy by using support automation tools in their profession. By self-exemplifying features and tools in the Adobe products, the participants derived many of which could be related to software automation (Software automation in the sense of relieving the user from certain tasks). All the participants except one stated that they use automation tools (Mainly related to the Adobe products). The most utilized and by far most convenient ones were the geometrical automation tools. Examples of these tools that were mentioned are tools of the transform panel of Adobe and the alignment panel of Adobe, guides and grids etc. The automation aspect in these tools are the automatic alignment of objects, the automatic regular distribution of objects, automatic snap to guide and / or grid etc. More powerful and possibly vendor-specific tools / features mentioned were “step and repeat”, “layer composition”, “slicing”, text-wrapping (General vendor-independent functionality), pagination (General vendor-independent functionality) etc.

5.2 Practitioner Evaluation of Generated Layouts

The study survey aimed to establish knowledge related to research question 1. The survey consisted of document layouts generated by the prototype layout automation system LaG that the participants were to evaluate. These layouts had been evaluated by the LaG system (Clearly, since LaG is based on evaluation and constraint solving techniques) preceding the survey session.

5.2.1 System Evaluation vs Practitioner Evaluation

The system evaluation data and the practitioner evaluation data was statistically compared to reveal whether or not the evaluation made by the system predicts the perceived layout quality evaluated by the participants. The data were compared based on several statistical tools, none of which indicating significant correlation between the system evaluations and the participant

<i>Pearson correlation</i>	Overall eval. LaG	Sum participant attr.eval.	Overall eval. participant
Overall eval. LaG	1.000	.173	.177
Sum participant attr.eval.	.173	1.000	** .960
Overall eval. participant	.177	** .960	1.000

Table 3: System overall evaluation vs participant overall evaluation

<i>Pearson correlation</i>	LaG balance	LaG alignment	LaG equilibrium
<i>Participant evaluations</i>			
Overall	-.046	.195	.192
Balance	.017	.196	.192
Horizontal balance	-.103	.241	.245
Vertical balance	-.025	.018	.011
Symmetry	-.007	.040	.034
Alignment	-.129	.265	.258
Equilibrium	.356	.128	.126
Efficiency	.300	.165	.156

Table 4: System evaluation vs participant evaluation

evaluations.

The system's overall quality measure of a document layout is based on a function of the weighted attribute evaluations. The participants make a specific overall evaluation in addition to attribute specific evaluations. Table¹² 3 and table 4 indicates that there are no correlation between the overall evaluation of the system and the overall evaluation of the participants. Neither if we compare system evaluation to the overall participant evaluation nor when we compare the system evaluation to the sum of the participants attribute evaluations (Which also gives an overall-type of score).

5.2.2 Practitioner Evaluation

The thesis aim is not to problematize practitioner evaluation in experimental settings related to quality measurement of visual phenomena, but it could be useful to take a glance at the interrelation between participant attribute evaluation to determine result robustness or reveal potential methodology weaknesses and data anomalies. Table 4 shows both the relation between system attribute evaluations and participant attribute evaluations. We have already described the results of comparing system evaluation versus participant evaluation. We find the interrelation between attribute evaluations intriguing; there are definite correlation between most of the attributes evaluated. In addition the participants overall evaluations correlate with the participants specific attribute evaluations. The most apparent correlation is between the evaluations of balance, symmetry and overall score. (Which confirm the weighting metric of Ngo where both symmetry and balance is weighted high due to it's high affect on overall visual quality [5])

Although the background variables are of minimal practical use in analyzing the research questions of this thesis (Due to the limited number of participants), they may be of interest related to the study implementation. Table 6 shows the relations between background variables and overall participant evaluation score. We can observe that there is negative correlation between experience and the overall score. This may indicate that experienced practitioners are more critical to what can be considered quality related to layout.

Another interesting aspect related to table 6 is the correlation between evaluation id and overall evaluation. The evaluation id is actually sequence generated by the RDBMS. A correlation between the evaluation id and the overall evaluation indicates that the participants evaluate the document layouts higher the further in the sample set they have come.

5.2.3 System Evaluation

The prototype document layout automation system used for the study were set to generate two types of document layouts. One mode where it generates layouts based on existing technology with an implemented balance operator (Balance evaluation manager)[9] and a second mode where it generates layouts based on an the extended set of attribute operators including balance, alignment and equilibrium. Table 8 shows there are no significant correlation between participants overall evaluation of basic system implementation compared to extended system implementation. Although the system with the extended set of operators scores slightly better on overall evaluation and the specific attribute evaluations, the results aren't unambiguous en-

^{1**} Correlation is significant at the 0.01 level

^{2*} Correlation is significant at the 0.05 level

<i>Pearson correlation</i>	Overall	Balance	Horizontal balance	Vertical balance
Overall	1.000	** .883	** .912	** .844
Balance	** .883	1.000	** .823	** .866
Horizontal balance	** .912	** .823	1.000	** .805
Vertical balance	** .844	** .866	** .805	1.000
Symmetry	** .876	** .847	** .780	** .856
Alignment	** .716	** .587	** .587	** .632
Equilibrium	** .605	** .516	** .543	* .445
Efficiency	** .700	** .583	** .542	** .565

<i>Pearson correlation</i>	Symmetry	Alignment	Equilibrium	Efficiency
Overall	** .876	** .716	** .605	** .700
Balance	** .847	** .587	** .516	** .583
Horizontal balance	** .780	** .587	** .543	** .542
Vertical balance	** .856	** .632	* .445	** .565
Symmetry	1.000	** .672	** .526	** .576
Alignment	** .672	1.000	.199	* .424
Equilibrium	** .526	.199	1.000	** .732
Efficiency	** .576	* .424	** .732	1.000

Table 5: Participant evaluation vs participant evaluation

<i>Pearson correlation</i>	Experience	Evaluation id	Sum part. attr. eval.	Overall LaG
Experience	1.000	**-.783	**-.429	-.056
Evaluation id	**-.783	1.000	** .509	.043
Sum part. attr. eval.	**-.429	** .509	1.000	.061
Overall LaG	-.056	.043	.061	1.000

Table 6: Background variables vs system and participant evaluations

ough to conclude in the direction of system improvement. The evaluation score of the attribute alignment shows the largest difference, but still within the standard deviation interval. The evaluation score of the attribute equilibrium shows no significant improvement between the two implemented system modes.

Another aspect that is interesting in the scope of the system evaluation is that there is a slight improvement in participant balance evaluation from the basic implementation to the extended implementation according to table 7. We have to stress the fact that this indication is not significant, but it is interesting that the balance evaluation isn't deteriorated by the use of an extended operator set (Table 8).

	Mode	N	Mean	Std.deviation	Std.error mean
Balance	Basic	83	3.51	2.281	.250
	Extended	55	3.91	2.312	.312

Table 7: Evaluation of balance of samples generated in the two system modes

5.3 Proof of Concept System

The proof of concept layout automation system used in the survey also served a second purpose; a proof of concept of a service oriented document format independent layout automation system.

5.4 Results Related to Hypothesis 1

Hypothesis 1 is formulated as "Layout automation system utilization of additional operators for visual parameters strengthens the perceived visual quality of generated document layout. Operationalizing alignment and equilibrium in layout generation increases the perceived visual quality of generated document layout." The data from the survey where obviously used to confirm or discard this hypothesis. Based on the statistics of table 8 and figure 27 there is no indication that a layout automation system implemented with an extended set of preference attribute operators

immediately produces significantly better document layouts. There is no significant indication of that extending a layout automation system with alignment and equilibrium operators immediately increases the perceived visual quality of generated layout.

5.5 Results Related to Hypothesis 2

The proof of concept web based research framework is based on the implementation of the layout automation system LaG. LaG has an architectural structure derived from the proposed document format-independent layout automation system model. This means that we have showed that the system model is applicable to a real-world scenario context.

	Mode	N	Mean	Std.deviation	Std.error mean
LaG overall	Basic	15	.3419	.06923	.01787
	Extended	10	1.2243	.26591	.08409
Sum part. attr. eval.	Basic	15	125.8000	37.82327	9.76593
	Extended	10	135.8000	24.63872	7.79145
Participant overall	Basic	15	19.2667	5.96977	1.54139
	Extended	10	21.1000	3.14289	.99387
Balance	Basic	15	19.4000	6.93645	1.79098
	Extended	10	21.5000	5.23344	1.65496
Horizontal balance	Basic	15	17.4000	5.43533	1.40340
	Extended	10	19.8000	3.58391	1.13333
Vertical balance	Basic	15	17.4000	6.90548	1.78299
	Extended	10	17.6000	4.81202	1.52169
Symmetry	Basic	15	17.4667	8.03445	2.07449
	Extended	10	18.0000	4.71405	1.49071
Alignment	Basic	15	16.8667	6.31174	1.62960
	Extended	10	20.4000	4.81202	1.52169
Equilibrium	Basic	15	19.1333	5.79244	1.49560
	Extended	10	19.7000	5.20790	1.64688
Efficiency	Basic	15	18.1333	4.56488	1.17865
	Extended	10	18.8000	4.44222	1.40475

Table 8: Total score (Sum of all participants' attribute evaluations) of basic implementation vs extended implementation for every layout

6 Analysis

6.1 General

6.2 Review of Results

The results from the statistical analysis of the data retrieved in the study gave ambiguous results regarding research question 1 and hypothesis 1. Not in the sense of statistical ambiguity directly related to the hypothesis, but rather a possible ambiguity related to the validity of the data and potential bias the results. The analysis of the data itself discards hypothesis 1 (Table 8 and table 3) as it shows no significant qualitative improvement of document layout generated from the basic system to layout generated from the extended system. We are not trying to undermine these findings, but results related to the methodological approach are worth a proper discussion.

First of all the results indicate that there is no immediate qualitative improvement in document layout generated with a layout automation system based on a balance attribute operator compared to a system based on an extended set of attribute operators. This means that extending system layout evaluation with additional attribute operators has no immediate layout quality predictive value (Fig. 27 and 28). These findings can't be generalized to every kind of attribute operator, but rather to the specific operators in the extended attribute set presented. These findings can neither be generalized to any kind of attribute operator quantification metric, but to the metrics used in the prototype system.

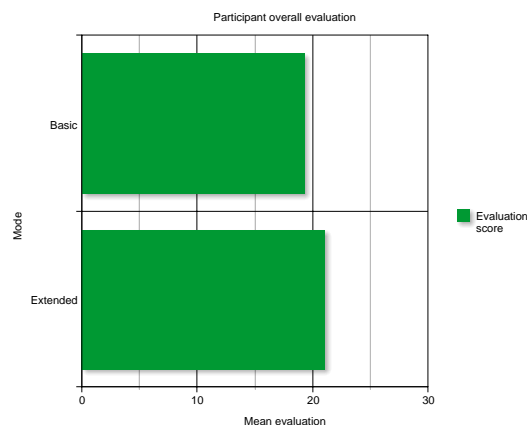


Figure 27: Participant overall evaluation of the two modes tested

The predicament of these findings is the results revealed when analyzing the participants evaluation pattern. Table 5 and table 6 shows three interesting tendencies related to participant evaluation; inter-attribute correlation, correlation between attributes and overall evaluation and correlation between evaluation id and overall evaluation.

The inter-attribute correlation indicates that whenever the participants are rating a certain attribute high significantly predicts high rating of other attributes as well. The same holds for overall rating which predicts the attribute evaluation and vice versa. So why are these findings interesting related to the thesis research problem? The internal and external validity of experiments related to subjective layout quality and layout characteristic measures depends on how the participants act in experimental settings. The data presented related to these aspects may

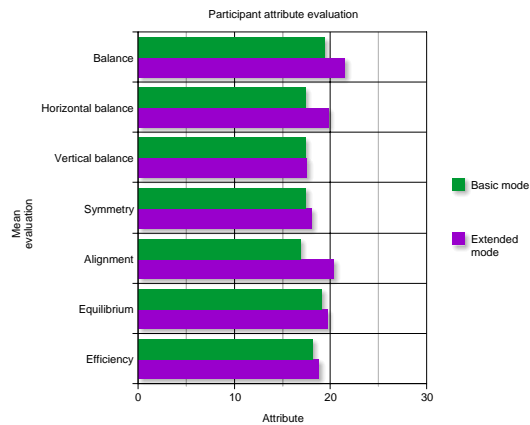


Figure 28: Participant attribute evaluation of the two modes tested

indicate participant evaluation ambiguity which can harm the internal validity.

Research question 2 and hypothesis 2 are answered by the literature study provided and by the technological testing of the layout automation framework used in the study. The LaG system is a proof of concept system modeled based on the modeling framework to prove a real-world implementation of a flexible document format-independent layout automation system. The proof of concept system is implemented and set up as a distributed system publicly available for utilization by a wide range of document layout application. The modeling framework provides modeling flexibility regarding architectural modeling and technological utilization. Typical layout automation systems based on evaluation techniques can easily be modeled after the modeling paradigms adapted and implemented in the modeling framework. The actual real-world utilization as an experimental tool in a research framework proves the flexibility of the LaG system and thereby the flexibility of the modeling framework.

6.2.1 Research Question 1

Let us review the actual statistical data presented in the “Results”-chapter. The most important results related to this specific research question can be viewed in table 8 and table 7. Based on the numbers presented there is no significant improvement of generated layout quality from the basic LaG system to the extended LaG system. There may be several reasons for this. The technological implementation of a layout automation system is generally complex and many typical components affects the overall quality of the layout generated. First of all we need to stress that the most likely explanation is that the layout automation system really doesn’t produce better quality layout when using an extended set of attribute operators. But there are some other aspects we need to problematize before concluding.

A potential technological explanation for the results retrieved is system inadequate technological implementation; either related to this particular study or related to the technological advance of the research area. Off course, this thesis wouldn’t have it’s right to exist if it weren’t for the current inadequacy of evaluation techniques related to document layout automation. A more interesting aspect is the inadequacy of the system implementation of the particular system

LaG. Is the metrics used sufficient to operationalize the phenomena we're working with? Is the quantification made by the programmatic implementation of the metric sufficient? In general the metric used are based on existing research and should not generate result bias. [11] [5] A programmatic implementation based on the metrics presented by Harrington et al shouldn't be a problem as the procedures are relatively straight forward. Still such a programmatic implementation of the attribute operators alignment and equilibrium hasn't been extensively tested before as far as we can see. A further exploration of practical implementation of such metrics might be of interest in the future.

Another aspect related to the internal validity of the results is how participants evaluate layout. Based on the statistics of table 5 one can recognize an evaluation pattern. We see a distinguished inter-attribute evaluation correlation and correlation between specific attribute evaluation and overall evaluation. This means that the participants have a tendency to rate the specific attributes higher the higher they rate the overall layout. Although the number of practitioners used in the study denies an generalization of the phenomenon, it is interesting to review it. The fact that these practitioners all in all have made about 150 evaluations of document layouts indicates that there is a tendency. This really not that surprising. One might argue that the practitioners have a top-down approach to the practice of evaluating design aesthetics. The overall impression of a layout may be far more important than the individual attributes. We state "may" as the collected data from this study aren't robust enough to fundament conclusions related to this phenomenon as they're retrieved to answer completely different questions. But these aspects are entirely related to the internal validity of the study and are therefore of a great deal of interest in a methodological perspective and can explain potential result bias.

Considering the potential bias related to participant evaluation it is interesting to view the results of the specific attribute evaluations related to the two system implementations used. If we take another glance at the results of table 8 we see that all the attributes shows improvement from basic implementation to extended implementation, although not significantly. Alignment is the attribute which have improved the most, actually on the verge of improving significantly. One might reason that this improvement would have been confirmed if the participant evaluation bias where counterbalanced as the specific attribute evaluations would profit from this. This isn't entirely true; we have no data to support this assumption and it needs to be further investigated. Another explanation of the slight improvement of the evaluations depending on system implementation might be technological. The metrics and the programmatic implementation may be working, but not sufficiently. Further practical testing of specific metrics and the implementation of the metrics should be of interest in future research.

6.2.2 Research Question 2

Research question 2 is related to the development of a layout automation system model framework for document format-independent layout automation systems. The implementation of the layout automation system LaG serves as a proof of concept related to research question 2. The system is based on a system model framework which enables efficient and reliable layout automation system modeling. The LaG system is implemented as a distributed service publicly available for utilization by any document layout application. We have also developed a prototype appli-

cation utilizing the LaG system. A web application practically used in the experimental research framework of this research project. The web application requests a document layout through a web document layout generator API. The API uses LaG to generate the layout, maps the document format-independent model to HTML and sends it back to the application as a response. The system model pattern makes it extensible related to document generation interfaces (API may be implemented for any type of document format) and thereby document format-independent.

Although the system modeling framework is consistent and utilizable related to layout automation system modeling, the framework-based implementation of the LaG system is improvable in many ways. There is a need for extensions related to the document formats it supports. In the current prototype these extensions needs to be implemented as separate API's, an architecture that should be reviewed. We suggest the use of standardized communication format technologies such as SOAP in a larger extent. In this perspective there is a demand for a document technology independent layout description language for the standardization of this communication. An example of such an extension is the use of SOAP for communicating an ontology-based (Standardized terminology and annotation of semantics) XML-representation of the document layout. Another approach would be to test the utilization of CORBA, which is already implemented through Java RMI but haven't been tested.

6.2.3 System Evaluation

The internal system evaluation of document layout attributes are essential to how the system performs. The system is built upon utilizing evaluation techniques for the layout quality optimization. In a system perspective it is interesting to analyze data retrieved from the system attribute evaluations. It seems as though the extended set of operators doesn't make the layout automation system perform adequately. We have already discussed possible explanations for this in the perspective of the research problem. In a system perspective it is more interesting to reflect upon how the system evaluators affect each other. The two main components of the implemented automation system is the evaluator (With the attribute evaluation managers) and the constraint solver. These components have huge effect on the layout evaluations upon which the layout generator optimizes the layout quality. The evaluator have direct effect as it quantifies the notion of quality. This quantification is the variable on which the layout generator optimizes. The constraint-solver have an indirect effect as it resolves aspects of layout unsuitable for evaluation such as overlapping elements, margins and keeping elements within document surface borders.

The use of several attribute operators in a layout automation system may have unforeseen effects. A layout automation system basing it's evaluation on one specific attribute is specialized on optimizing in perspective of that specific attribute. The use of an extended set of operators moderates the specific attributes effect on the overall evaluation; the characteristics of the specific attributes are deteriorated for the sake of the overall layout quality. Although there have been presented regression analyzes and metrics for proper weighting of these attributes [4] [9] [10] [11] [5], there is no research projects discussing the practical implementations of the attribute metrics as far as we can see. We have discovered that the implementation of metrics using an extended set of attribute operators is no guarantee for improving general layout quality. The reasons may be that the operators within a metric context is challenging each others prominence,

resulting in overall layout quality more on the average rather than improving. The weighting metric used in LaG resulted in a correlation between system overall layout evaluation and the specific system attribute evaluations. These results do either confirm the presented theory or indicate a faulty weighting metric, both of which motivates a further investigation of inter-attribute effect in document layout automation evaluation techniques.

6.2.4 System Evaluation vs Participant Evaluation

The results from the study implemented in this thesis is interesting related to the study design of future research. There are aspects of the methodological and experimental setting in this research project that needs to be further investigated. The aspect of participant overall evaluation predicting attribute evaluation has already been discussed. Another interesting element related to the study design is how participants evaluate during the course of the survey session. Looking at table 6 we actually see that there is a correlation between evaluation id and sum of attribute evaluations (Which corresponds to overall evaluation). At first we considered this correlation had occurred accidentally. But when reflecting upon the fact that the evaluation id is sequence generated it actually hasn't happened by chance. This correlation is an indication that the evaluation id predicts the participant overall evaluation; the further a participant has come in a sample set, the higher he / she ranks the samples. It seems as though the participants calibrate their evaluations during the survey session. This may reason in that the participants have high expectations to the samples they are to be presented prior to the survey, and adapts to reality when facing the actual quality of the samples. Another possible explanation is participant fatigue during an cognitively exhaustive survey session (A typical session lasted for 45-60 minutes).

These two (Inter-attribute evaluation correlation and evaluation calibration) methodological aspects of layout automation research are of utmost interest for the future research in the field. The research methodology used in this thesis is inspired by the methodological approaches used in earlier research in the field. If these experimental phenomena is generalizable to these kind of research projects it can harm the internal validity of the collected data. The statistical analysis of these problems is not the scope of this thesis and therefore we do not have the data to explicitly confirm methodological weaknesses.

6.3 Summary and Conclusions

The principle of document layout automation has a large potential related to the efficacy of layout production workflow. This thesis provide a deep dive into both the general utilization of document layout automation techniques and the technologies used for systems handling this kind of automation. The utilization of document layout automation varies in a wide range of applications. The designer might use layout automation tools to ease the work tasks related to the design process; tools automatically suggesting a base layout, tools laying out or constraining certain graphical elements etc. There are also applications related to distribution of information that it is practically impossible to offer individual layout; online publications might offer individual presentation of information content, automatic generation of newspaper articles based on content and individual preference etc.

Two main perspectives have been investigated in this thesis; the utilization of preference

attribute operators in layout automation and general modeling of layout automation systems. Here we will summarize the findings of this research project and try to draw some conclusions related to the analysis of the results retrieved in the study. In addition we will present some research problems suitable for future research based on the experiences we have made during the course of this project.

6.3.1 Hypothesis 1

Let us revisit hypothesis 1: “Layout automation system utilization of additional operators for visual parameters strengthens the perceived visual quality of generated document layout. Operationalizing alignment and equilibrium in layout generation increases the perceived visual quality of generated document layout”.

Based on the results of the survey this hypothesis has not been confirmed. Although the results indicate improvement from the basic implemented system to the extended system, these indications aren’t significant to conclude external validity. Some bias have been found related to the result based on the study design and participant’s evaluation of layouts, but these can’t be confirmed to have effect on the outcome unless they are further investigated.

6.3.2 Hypothesis 2

Let us revisit hypothesis 2: “Does a layout automation system model framework for document format-independent layout generation prove to be realistic in a real-world implementation?”. The thesis presents a practical implementation of a layout automation system to prove the generalities of layout automation system modeling.

The LaG system is based on several important general components and features making it flexible regarding document format generation and technology utilization. LaG is implemented as a service-oriented layout generator. LaG communicates through a web application API; a web application interface and “document model to web document”-mapping tool. The web application used in the research framework communicates with LaG through the API; a practical implementation of a web application utilizing the power of the LaG system (Uses the web application API as communication interface). The LaG-system’s architecture is based on the layout automation system model framework presented in this thesis. The actual practical application of the LaG system in the research framework proves the system model frameworks practical application in a real-world scenario. It further proves the sufficient technology and modeling flexibility a modeling framework ought to have to ensure application in a wide variety of modeling scenarios.

6.3.3 Conclusions

The utilization of visual preference attribute operators in layout automation techniques is a general technique implemented in a wide range of research related prototype systems in the field of layout automation. We have investigated the relationship between the use of attribute operators and generated layout quality. The results of the study described in this thesis doesn’t indicate an immediate effect of extending the layout attribute operator set related to the produced layout quality. The results aren’t generalizable for specific preference attribute operators in layout automation systems other than the operators implemented in this project. A layout automation system based on evaluation techniques extended with the attribute operators alignment and

equilibrium doesn't necessarily generate better quality layouts than a layout automation system merely implemented with a balance operator.

The system model presented in this thesis is proven as a model framework for document format-independent layout automation systems applicable in real-world scenarios. This indicates that general layout automation systems may use this model framework as inspiration for the system architecture modeling. The LaG system architecture was efficiently utilized as a research framework used in the study of this research project. The LaG-based research framework provided an essential proof of concept system for layout generation testing.

Preference Attribute Operators

Based on the analysis of the results retrieved in this research project we can derive several issues suitable for further investigation. Although the study shows that there are no immediate relationships between an extended attribute operator set and perceived layout quality, this is far from conclusive for the general use of preference attribute operators. This thesis is limited to the investigation of the preference attributes balance, alignment and equilibrium, three of several others defined in earlier research [11] [4] [5].

Layout quality isn't directly determined by the presence of specific characteristics, but rather a complex phenomenon which isn't straight forward operationalizable by mathematical description. The inter-attribute effect of the overall layout quality is a field mature for further investigation. How does one specific attribute affect other attributes? How does the constellation of attributes affect each other? More importantly, how does a specific attribute affect the overall layout quality? In the perspective of evaluation techniques in layout automation systems, both the study of inter-attribute relationships and the study of specific attribute effect are of importance for further technological advance.

Participant Evaluation

The statistical analysis of the data retrieved in the study revealed some surprising and interesting tendencies related to participant evaluation. The results indicated that there is correlation between participant's overall evaluation of layout quality and evaluation of specific attributes. It seems as though the participants may have a "top-down" approach to evaluating layout quality as the overall evaluation affects how they evaluate the specific attributes. Due to high degree of creativity related to the process of designing graphical products, this isn't as abnormal as one might think. A design is more defined by its totality than by its individual characteristics. A perfectly balanced layout helps the totality of the layout, but doesn't determine it. Design quality is general a product of its global preference characteristics and the perfect balance between these characteristics. This has actually been confirmed by some of the participants after the survey. The participants find it hard to distinguish and evaluate specific characteristics (Such as balance, alignment etc.) of layout.

Another tendency revealed by the data of the study related to participant evaluations is the fact that the quality evaluation changes during the course of a survey session. The participants actually rates layout quality higher the further in the sample set they have come. This is merely a tendency in the data set, not a basis for proven facts. It seems as though the participants experience some kind of survey fatigue during the session. Either the participants are actually

exhausted by a long lasting and concentration intensive survey or the participants are calibrating to the quality of the sample set presented.

A third interesting, but not as surprising, tendency is the negative correlation between evaluation and practitioner's experience. The experienced participant have a tendency to rank layout quality lower. This may be explained by the practitioner being more particular as they gain experience and a trained eye.

We have to precise that the tendencies presented in this section isn't proven as more than what they are; tendencies. But tendencies are often suitable for further investigation. These three phenomenon may be of research interest related to visual aesthetic quality experiments.

System Evaluation

The LaG system uses layout automation evaluation techniques for layout quality optimization. The metrics used for evaluation of preference attributes are inspired by diverse earlier research [9] [11] [5]. There are a few aspects related to the system implementation that is suitable for future research. As we've already described, the use of preference attribute operators in layout automation is a complex matter. In this perspective it would have been interesting to investigate the existing metrics for attribute operationalizing in real-world scenarios. Ngo, Harrington et al, Bauerly and Liu, Lok et al and Ahmad presents metrics related to attribute operationalization. Some of them have already been implemented, some are still only mathematically described. Another important research problem is the practical implementation of suitable attribute set weighting metrics. The system overall evaluation is a function of the attribute evaluations. To establish a working metric the inter-attribute relationships must be extensively mapped.

Bibliography

- [1] Sun. 2009. Java ee. <http://java.sun.com/javaee/>.
- [2] Kelly, K. 1998. *New rules of the new economy*.
- [3] Purvis, L., Harrington, S., O'Sullivan, B., & Freuder, E. C. 2003. Creating personalized documents: An optimization approach. *DocEng*.
- [4] Lok, S. & Feiner, S. 2001. A survey of automated layout techniques for information presentations. In *SmartGraphics*.
- [5] Ngo, D. C. L. 2000. Measuring the aesthetic elements of screen designs. *Displays*, 22, 73–78.
- [6] Bauerly, M. & Liu, Y. 2006. Computational modeling and experimental investigation of effects of compositional elements on interface and design aesthetics. *International Journal of Human-Computer Studies*, 64, 670–682.
- [7] Obrador, P. 2007. Document layout and color driven image retrieval. *Sigir*.
- [8] Faria, A. C. & de Oliveira, J. B. S. 2006. Measuring aesthetic distance between document templates and instances. *DocEng*.
- [9] Lok, S., Feiner, S., & Ngai, G. 2004. Evaluation of visual balance for automated layout. *IUI*, 13–16.
- [10] Ahmad, A.-R. *An Intelligent Expert System for Decision Analysis and Support in Multi-Attribute Layout Optimization*. PhD thesis, 2005.
- [11] Harrington, S. J., Naveda, J. F., Jones, R. P., Roetling, P., & Thakkar, N. 2004. Aesthetic measures for automated document layout. *DocEng*, October, 28–30.
- [12] Ivory, M. Y. & Megraw, R. 2005. Evolution of web site design patterns. *ACM Transactions on Information Systems*, 23, 463–497.
- [13] Liestøl, G. & Rasmussen, T. 2007. *Digitale medier - En innføring ("Digital media - an introduction")*. Universitetsforlaget.
- [14] Syvertsen, T. 2004. *Mediemangfold ("Media plurality")*. IJ forlaget.
- [15] Zhang, X. & Ren, X. 2006. Cater: A framework for the automated layout of transactional pages. *Proceedings of The Sixth IEEE International Conference on Computer and Information Technology*.

- [16] González, J. & Merelo, J. J. 1998. Optimizing web page layout using an annealed genetic algorithm as client-side script. *PPSN*, 1018–1027.
- [17] Badros, G. J., Borning, A., Marriott, K., & Stuckey, P. 1999. Constraint cascading style sheets for the web. *UIST*.
- [18] González, J., Rojas, I., Pomares, H., Salemerón, M., Prieto, A., & Merelo, J. 2001. Optimization of web newspaper layout in real time. *Computer Networks*, 36, 311–321.
- [19] Goldberg, J. H., Helfman, J. I., & Martin, L. 2008. Information distance and orientation in liquid layout. *CHI*.
- [20] Kleinberg, J. & Tardos, E. 2006. *Algorithm Design*. Addison - Wesley.
- [21] de Oliveira, J. B. S. 2008. Two algorithms for automatic document page layout. *DocEng*.
- [22] W3C. 2009. Xhtml 1.0, css 3. <http://www.w3c.org>.
- [23] Jacobs, C., Li, W., Schrier, E., Bargeron, D., & Salesin, D. 2003. Adaptive grid-based document layout.
- [24] Bateman, J., Kamps, T., Kleinz, J., & Reichenberger, K. 2001. Towards constructive text, diagram and layout generation for information presentation. *Association for Computational Linguistics*, 27.
- [25] Cai, S., Yu, J., & Xu, Y. 2008. The effects of web site aesthetics and shopping task on consumer online purchasing behavior. *CHI*.
- [26] Xie, X., Wang, C., Chen, L.-Q., & Ma, W.-Y. 2005. An adaptive web page layout structure for small devices. *Multimedia Systems*, 11, 34–44.
- [27] Ishak, E. W. & Feiner, S. K. 2007. Content-aware layout. *CHI*.
- [28] Badros, G. J., Borning, A., & Stuckley, P. J. 2001. The cassowary linear arithmetic constraint solving algorithm. *ACM Transactions on Computer-Human Interaction*, 8, 267–306.
- [29] Johari, R., Marks, J., Partovi, A., & Shieber, S. 1997. Automatic yellow-pages paginatino and layout. *Journal of Heuristics*, 2, 321–342.
- [30] Kamba, T., Bharat, K., & Albers, M. C. The krakatoa chronicle - an interactive, personalized, newspaper on the web.
- [31] Fogarty, J. & Hudson, S. E. 2003. Gadget: A toolkit for optimization-based approaches to interface and display generation. *ACM*.
- [32] Bruno, L. 2008. An interface for your eyes only. <http://www.guardian.co.uk/technology/2008/aug/28/research.artificialintelligenceai>.
- [33] Gajos, K. 2009. Krzysztof gajos home page. <http://www.eecs.harvard.edu/kgajos/>.

- [34] Gajos, K. & Weld, D. S. 2004. Supple: Automatically generating user interfaces. *IUI*, 1, 13–16.
- [35] Kröner, A. 1999. The designcomposer: Context-based automated layout for the internet. *American Association for Artificial Intelligence*.
- [36] Baraglia, R. & Silvestri, F. 2007. Dynamic personalization of web sites without user intervention. *Communications of the ACM*, 50.
- [37] Nasraoui, O. & Petenes, C. Combining web usage mining and fuzzy inference for website personalization.
- [38] Lohmann, S., Kaltz, J. W., & Ziegler, J. Dynamic generation of context-adaptive web user interfaces through model interpretation.
- [39] Hosobe, H. 2005. Solving linear and one-way constraints for web document layout. *ACM Symposium on Applied Computing*.
- [40] Kim, Y., Gotoh, K., & Toyosada, M. 2003. Automatic two-dimensional layout using a rule-based heuristic algorithm. *Marine Science and Technology*, 8, 37–46.
- [41] Luger, G. F. & Stubblefield, W. A. 1998. *Artificial intelligence*. Addison - Wesley.
- [42] Angeli, A. D., Sutcliffe, A., & Hartmann, J. 2006. Interaction, usability and aesthetics: What influences users' preferences. *DIS*.
- [43] Leedy, P. D. & Ormod, J. E. 2005. *Practical research - Planning and design*. Pearson Merrill Prentice Hall.
- [44] Harold, E. R. & Means, W. S. 2004. *XML in a nutshell*. O'Reilly.
- [45] Bradley, N. 2002. *The XML companion*. Addison - Wesley.
- [46] Møller, A. & Schwartzbach, M. 2006. *An Introduction to XML and Web Technologies*. Addison - Wesley.
- [47] Pressman, R. S. 2001. *Software engineering - A practitioner's approach*. McGraw Hill.
- [48] Bell, J. T., Lambros, J. T., & Ng, S. 2003. *J2EE Open Source Toolkit - Building an Enterprise Platform with Open Source Tools*. Wiley.
- [49] JBoss. 2009. Jboss community. <http://www.jboss.org>.
- [50] Center for Geometric Computing, B. U. 2009. The data structures library in java (jdsl). <http://www.jdsl.org/>.
- [51] Kipphan, H. 2001. *Handbook of Print Media*. Springer.
- [52] Adobe. 2009. Adobe postscript 3 and pdf. <http://www.adobe.com>.

- [53] Bruno, V., Tam, A., & Thom, J. 2005. Characteristics of web applications that affect usability: A review. *OZCGHI*.
- [54] Badros, G. J. & Borning, A. 2009. Cassowary constraint solving toolkit. <http://sourceforge.net/projects/cassowary/>.
- [55] Badros, G. J. & Borning, A. 2005. Cassowary constraint solving toolkit. <http://www.cs.washington.edu/research/constraints/cassowary/>.
- [56] Sun. 2009. Java. <http://java.sun.com/>.
- [57] PostgreSQL. 2009. Postgresql. <http://www.postgresql.org/>.
- [58] Hibernate. 2009. Hibernate. <https://www.hibernate.org/>.
- [59] JQuery. 2009. JQuery. <http://jquery.com/>.
- [60] 2009. *The American Heritage Dictionary of the English Language, Fourth Edition*. Houghton Mifflin Company.
- [61] Fu, T. & Liu, M. 2004. A gateway from html to xml. *Proceedings of the International Database Engineering and Applications Symposium*.
- [62] Schlungbaum, E. 1997. Individual user interfaces and model-based user interface software tools. *ACM*.

A Technological implementation

A.1 Code

The java code implementation of the LaG system is publicly available from the following adress:
http://www.stud.hig.no/~001705/LaG/LaG_System.zip.

B Document Layout samples

B.1 Randomly generated samples

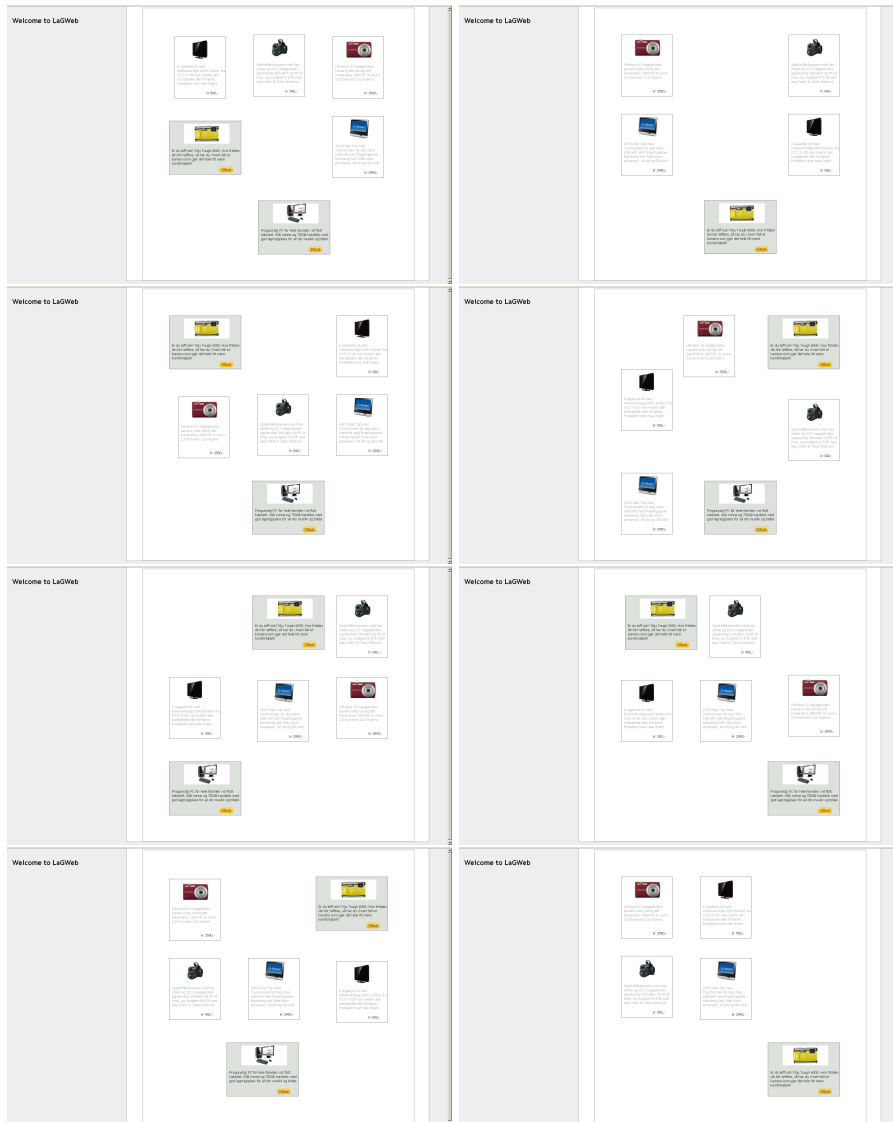


Figure 29: Samples generated by the prototype layout automation system LaG

B.2 WeightMap samples



Figure 30: WeightMaps generated by the prototype layout automation system LaG

C Survey User Interfaces

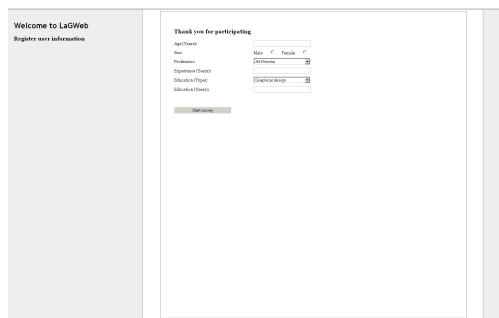


Figure 31: Initial survey user interface

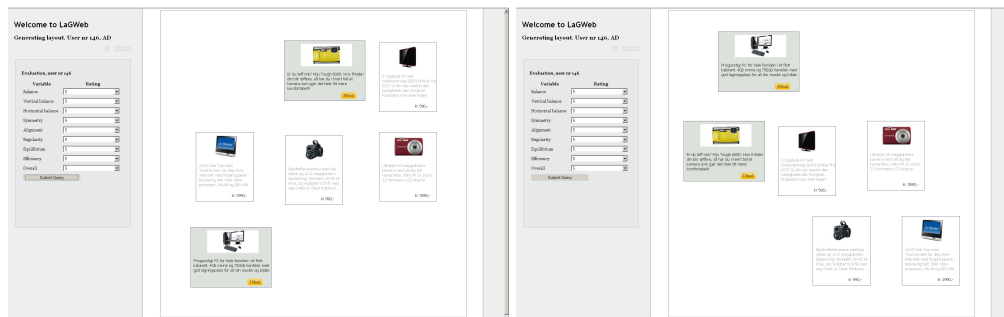


Figure 32: Survey evaluation user interfaces. Layout samples generated from basic system mode.

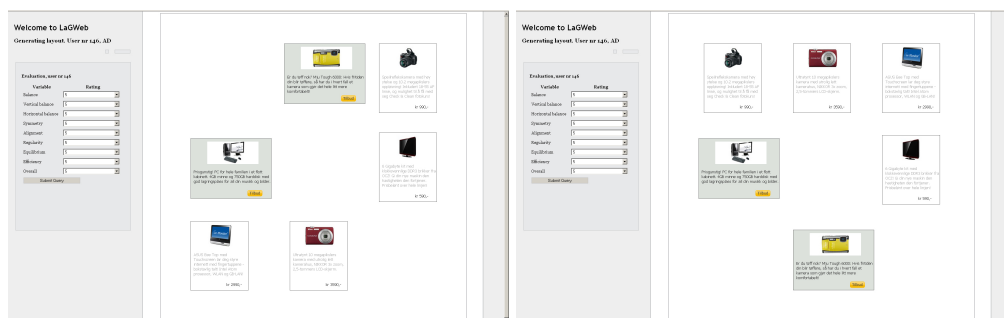


Figure 33: Survey evaluation user interfaces. Layout samples generated from extended system mode.

D Survey implementation

Interview questions

Effektivisering av arbeidsoppgaver innen layout-komposisjon *(Efficiating work tasks in the process of composing layout)*

Deltager nr

Jeg har et behov for å bruke verktøy som kan automatisere tidkrevende og repetitive arbeidsoppgaver i forhold til de gitte situasjonene under.

(I am dependent on tools to automatize tedious and repetitive tasks regarding the given situations below)

Jeg jobber ofte med layout hvor de grafiske elementene settes sammen med relativt like repetitive og gjentagende steg (Eksempelvis å sette opp en grafisk presentasjon av en produktkatalog, hvor alle produktene kategoriseres og settes opp etter en gitt mal). *(I often work with graphical design, where each graphical elements are layed out in a relatively repetitive manner.)*

Designoppdragene jeg er involvert i er designmessig preget av høy grad av kreativitet. *(Design tasks that I work with Is highly creative of nature)*

Jeg har et behov for å bruke verktøy som kan lette tidkrevende og repetitive oppgaver i forhold til layout-komposisjon. Se under for eksempler på slike typer oppgaver.

(I need automation tools that eases tedious and repetitive tasks in the process of composing layout.)

Eksempler

Generell gridbasert (Rutenett/modulsystem) layout. *(General grid based layout)*

Layout av forsiden av en skjønnlitterær bok. *(Layout of the front of a novel in book form.)*

Layout av artikler på hovedsiden til en nettavis. (*Layout of articles on the main page of a newspaper online.*)

Layout av en plakat i forbindelse med markedsføring av en musikkfestival. (*Layout of a poster of a music festival within a marketing context.*)

Layout av informasjon om treff på bøker i et søk gjort hos en bokhandel på nett. (*Layout information of hits regarding books in a search done at an online book store.*)

Layout av et stort antall annonser i en trykt publikasjon. (*Layout of a large number of ads in a printed publication.*)

Bruk av etablerte verktøy innen layout komposisjon

(The use of well-established tools in the process of composing layout)

Kommenter påstandene under. *(Evaluate the allegations under.)*

Jeg bruker hjelpemidler for å forenkle arbeidet (Grid-systemer, maler etc.) i forhold til layout med repetitive og gjentakende grafiske elementer. *(I use general tools (Grid systems, templates etc.) for easing layout composition in my profession.)*

Jeg bruker verktøy for å automatisere layout-komposisjon i min arbeidshverdag. *(I use automation tools for layout composition in my profession.)*

Hvis du sier deg enig i forrige påstand; hvilke verktøy bruker du for å automatisere prosessen i å sette opp layout? *(If you agree to the previous allegation; which tools do you use to automatize the process of composing layout?)*

D.1 Survey results, database dump

The results from the survey is publicly available from the following adress: http://www.stud.hig.no/~001705/LaG/layout_dump_with_headers.csv.