AALBORG UNIVERSITY

IT-supported Rostering:
An Investigation of Literature and IT-systems

MASTER THESIS
Information Systems (IS)
Department of
Computer Science

WRITTEN BY
Simon Sønderskov Iversen
Christian Sand Pilgaard
Daniel Ginnerup Kappers

SUPERVISOR
Jan Stage

Title:
IT-supported Rostering: An Investigation of Literature and IT-systems

Aalborg University
Department of Computer Science
Selma Lagerlöfs Vej 300
9220 Aalborg Øst
http://cs.aau.dk

Topic:
Information Systems (IS)

Project period:
February 1, 2016 – June 9, 2016

Semester:
10th Semester – Master Thesis

Group:
IS1010F16

Students:
Simon Sønderskov Iversen
Daniel Ginnerup Kappers
Christian Sand Pilgaard

Supervisor:
Jan Stage

Published:
June 9, 2016

Synopsis:

In this report, we set out to examine the subject of IT-supported rostering with the purpose of identifying what the literature presents on the subject, how this can be used to analyse existing systems, and finally how it can be used to explore new approaches to IT-supported rostering. This was done, by developing three research questions using three different research methods. In order to identify what the literature contains a literature review was conducted. Berrypicking was used to find 77 texts, through a series of techniques. Based on these, a framework with 3 categories and 13 parameters was created. Next, a Content Analysis of existing commercial rostering IT-systemswas conducted. A total of 46 systems were analysed using heuristics based on the parameters found in the first article. The results were three ways of describing the characteristics of the systems: An overview of constraints and how they are supported in rostering IT-systems, characteristics based on Rostering Approach, and clusters based on level of support for heuristics. Lastly, Basic Research was conducted in the form of a proof-of-concept, focusing on exploring different approaches for an IT-supported Self-rostering system in a complex rostering situation. A Human-Centred Design approach incorporating Effective Prototyping was used, to create three prototypes of Self-rostering systems. These prototypes were evaluated by a focus group, containing employees from a rostering practice with a complex rostering situation. Findings were not definitive; as it did not yield a definitive conclusion, but instead showed strong tendencies towards IT-supported Self-rostering being viable for a complex rostering situation.

Preface

This master thesis has been conducted in the spring of 2016, by project group IS1010f16 from Aalborg University, as a part of the Information Systems (IS) research unit.

This master thesis consists of three articles presented in the CHI format, which can be found in the appendix. The four chapters in this report focus on how these articles are relevant for IS as a research area.

Chapter 1 is an introduction, which includes the overall problem statement and the research questions for each article.

Chapter 2 describes the contributions of each article and how they are related to each other.

Chapter 3 describes the research method of each article, their strengths, weaknesses and which countermeasures were taken.

Chapter 4 conclude on each research question and the problem statement. Afterwards, limitations and future work is presented for each contribution.

The style of reference for this master thesis report is the 5th edition of APA, and the bibliography can be found in chapter 5.

We would like to thank the participants of the conducted interviews and the focus group for their participation and involvement. Furthermore, we would like to thank the companies for granting us access to the rostering IT-systems. Lastly, a thank you go to our supervisor Jan Stage for providing continuous feedback and assistance throughout the master thesis.

We hope you enjoy the report.

- IS1010F16

Abstract

This master thesis examines the subject of IT-supported rostering with the purpose of identifying what the literature presents on the subject, how this can be used to analyse existing systems, and finally how it can be used to explore new approaches to IT-supported rostering. This is accomplished through three CHI formatted articles, each covering their respective subject. The three articles' contributions, research methods and conclusions will be presented in this report. Prior to these presentations, an introduction, the research field, the research questions, and the problem statement, will be described.

The first of the three articles covers the existing literature on the subject of rostering, through an interdisciplinary Literature Review. It aims to find relevant parameters for IT-supported rostering situations. This is related to the problem statement of this report, as the parameters are necessary to identify, prior to an analysis of systems. The relevant literature was found through the Berrypicking method, where 11 texts were handpicked. Forward and backward search was performed upon these 11 texts, revealing a total of 1190 texts. The resulting amount was then screened through a 4 step screening process, using inclusion criteria, which resulted in a total of 77 texts. The texts were analysed using Open Coding and the resulting parameters were categorised using Affinity Diagramming. A total of 13 parameters were identified that were relevant to IT-rostering, and these were split into the three categories; Strategy, Context and Implementation.

The second article analyses existing IT-systems for rostering, with the purpose of giving insight into the distribution of the systems, related to how they support rostering. The article relates to the problem statement, as it takes different parameters from literature into consideration, and analyses a range of systems through heuristics created, based on these parameters. The systems were found through two comparison websites, where a total of 142 systems were found. The systems were then screened using inclusion criteria, resulting in 46 systems. These systems were evaluated using 26 heuristics, which was developed using the rostering parameters. The data was then analysed through descriptive statistics.

The third article focuses on the viability of IT-supported Self-rostering in a complex rostering situation, with the purpose of identifying tendencies or conclusions on the viability of a system. This relates to the problem statement, as the literature is very limited about IT-supported Self-rostering. Three prototypes are created through a Human-Centred Design process, where preliminary knowledge was gathered through two interviews. Afterwards three prototyping iterations using the Effective Prototyping method were conducted, followed by a final evaluation of the prototypes, through a focus group. The results revealed scepticism towards a system being able to implement the complexity of rules within their organisation, but found the concept of IT-supported Self-rostering advantageous, if the proper groundwork was put into the system. This shows a tendency towards IT-rostering being viable to implement in complex rostering situations.

Three studies were conducted, with different perspectives on IT-supported rostering. This is in order to answer the question of how existing IT-supported rostering systems can be analysed through parameters from the literature, and how this can contribute to the creation of new IT-supported rostering systems. It was managed to find core parameters for IT-supported rostering, analyse current commercial systems based on a selected range of the parameters, and create three prototypes to investigate different approaches to IT-supported Self-rostering in complex situations.

Table of contents

1 INTRODUCTION

Burke et al. (Burke, De Causmaecker, Petrovic, & Berghe, 2001, p. 1) writes: "*Employee rostering problems basically consist of assigning a number of tasks (or shifts) to personnel with different skills over a defined period of time*". While Burke et al. (Burke et al., 2001) makes some assumptions of what the rostering process includes, it helps define rostering as the act of assigning planning units to personnel, or vice versa, over a period of time. Rostering is an impactful activity prevalent in most organisations, which is often supported by IT (Cheang, Li, Lim, & Rodrigues, 2003; Higgins & Postma, 2004). The impact of optimising rostering practices through the use of IT can be found throughout the literature, where examples of greatly improved financial benefits can be found (Higgins & Postma, 2004), as well as optimisations of employee satisfaction and morale (Mason, Ryan, & Panton, 1998). Silvestro & Silvestro (Silvestro & Silvestro, 2008) emphasise the importance of fitting the organisational structure to the rostering approach, and presents three rostering approaches; Self-rostering, Team-rostering, and Departmental-rostering. They argue that each approach is best fitted to a certain Rostering Complexity and that there are a set of advantages and disadvantages related to each approach. To optimally create or change an IT-supported rostering situation, one has to include the different research areas and perspectives on rostering, and weigh the different advantages and disadvantages, against what one wants to accomplish.

As of spring 2016 there are many commercially available software solutions for rostering (Capterra Inc., ; Nubera, 2016). While many different IT-systems for rostering is presented through the academic literature (Burke, De Causmaecker, Berghe, & Van Landeghem, 2004), there are no available research focusing on commercially available systems in regards to their support rostering. So currently, there is no research to guide decisions when looking for commercial systems or if one has the desire to create a system that differs from the rest.

1.1 RESEARCH FIELD

Our research is conducted within the field of Information Systems (IS). Rostering is a time consuming and impactful activity most organisations are dependent upon. Therefore, IT-supported rostering has been subject to a great amount of research from different research areas with different perspectives. Our aim is to close the gap between academia and practitioners by synthesising knowledge of different research areas for practitioners to use. Furthermore, the aim is to examining at the current state of rostering systems and their support of rostering, and challenge the current approaches for creating rostering systems.

1.2 Problem statement & research questions

The aim of this master thesis is to study IT-supported rostering from three perspectives; literature, commercial systems and exploring new approaches. These perspectives have led to the following problem statement:

Problem statement: *How can existing IT-supported rostering systems be analysed using parameters found through literature, and how can this contribute to the creation of new IT-supported rostering systems?*

The problem statement is elaborated through the three research questions presented below:

1. Research question: *What parameters should be taken into account when creating IT-supported rostering practices, according to literature?*

The first research question focus on gathering knowledge about parameters for IT-supported rostering and determining which is relevant when creating an IT-supported rostering situation.

2. Research question: *How does existing commercial rostering IT-systems support rostering?*

The second research question focus on commercially available IT-systems for rostering and what characterises them. Knowledge from the previous research question will be used to categorise systems according to their support of selected rostering parameters.

3. Research question: *Is it possible to create a viable IT-supported Self-rostering system for a complex rostering situation?*

The third research question is an exploration of IT-supported Self-rostering in a complex rostering situation. We want to investigate the viability of IT-supported Self-rostering in a context that it would otherwise not be suited for. The focus will, however, not be on technical implementations of features and algorithms, but instead on the rostering process, and the interaction this entails.

These research questions are answered through three articles, one for each research question.

2 CONTRIBUTIONS

Chapter 2 presents the academically relevant contributions of our research, which is the result of our master thesis. This chapter consist of two parts; first part being an overview of the different contributions and the relation between them, and the second part being a summary of each article's individual contributions. It is recommended that the three articles are to be read sequentially, as the results of each article is used within the next.

2.1 OVERVIEW OF CONTRIBUTIONS

Our research contribution are the answers to our problem statement and research questions, described in the previous chapter. To answer the research questions, three articles we have been written, which comprises our research contribution:

Article 1: Iversen, S. S., Kappers, D. G. and Pilgaard, C. S. (2016). Parameters to Consider when IT-supporting Rostering: A Literature Interdisciplinary Review. Department of Computer Science, Aalborg University, Aalborg.

Article 2: Iversen, S. S., Kappers, D. G. and Pilgaard, C. S. (2016). A Review of Existing Commercial Rostering IT-systems. Department of Computer Science, Aalborg University, Aalborg.

Article 3: Iversen, S. S., Kappers, D. G. and Pilgaard, C. S. (2016). IT-supported Self-rostering in a Complex Rostering Situation. Department of Computer Science, Aalborg University, Aalborg.

The three articles can be found in Appendix.

The foundation for all articles are their relevance to the subject of IT-supported rostering. The focus on IT-supported rostering consist of two perspectives: Parameters for IT-supported rostering and IT-systems for rostering. In order to show the relation between the articles and their relation to the subjects, a 2x2 matrix has been constructed, as seen in Table 1 below.

	Existing	**New**
Parameters for IT-supported Rostering	Article 1	Article 1
IT-systems for Rostering	Article 2	Article 3

Table 1: Relationships between the three articles based on their focus.

The rows describe whether the article covers the subject of parameters for IT-supported rostering or IT-systems for rostering. The columns describe whether the articles cover the subject in relation to what already exists, or what new can be added to the area.

2.2 CONTRIBUTION 1

Iversen, S. S., Kappers, D. G. and Pilgaard, C. S. (2016). Parameters to Consider when IT-supporting Rostering: An Interdisciplinary Literature Review. Department of Computer Science, Aalborg University, Aalborg.

The contribution of article 1 is a framework for IT-supported rostering, intended for practitioners to use when creating or changing IT-supported rostering practices. The framework itself is based on an interdisciplinary Literature Review, scoped using Cooper's taxonomy for Literature Reviews (Cooper, 1985). As opposed to similar research, our framework includes all relations of the four dimensions of Leavitt's Diamond; People, Structure, Task, and Technology (Leavitt & March, 1962). In order to include the different elements from the perspectives of multiple research areas, Berrypicking (Bates, 1989) was used as a method to identify and select the foundation of the Literature Review. Footnote Chasing and Citation Searching was conducted on the selected texts, which resulted in a total amount of 1190 texts. Through a systematic screening process consisting of 5 steps, the 1190 texts were reduced to 77 texts. To analyse the 77 texts a process including Open Coding (Corbin & Strauss, 1990) and a table similar to a Concept Matrix (Webster & Watson, 2002) was used. It was decided to conceptualise and describe the complexity of IT-supported rostering through a framework, composed through a process inspired by Affinity Diagramming (Beyer & Holtzblatt, 1997).

The framework consists of three categories: Strategy, Context, and Implementation. The first category of the framework is Strategy, and concerns the overall goals and approach when changing or creating IT-supported rostering practices. The second category is Context, addressing: The setting of the rostering situation, how organisational structure influence the rostering situation, and what one can do to change it. The third and last category is Implementation, which is about the means of realising the strategy within the context. Each category of the framework and their underlying parameters, contribute to the understanding and use of the framework, which is explained through examples from the literature, in order to make it tangible for the practitioner. Every category and underlying parameters are interconnected and should be considered in relation to each other.

2.3 CONTRIBUTION 2

Iversen, S. S., Kappers, D. G. and Pilgaard, C. S. (2016). A Review of Existing Commercial Rostering IT-systems. Department of Computer Science, Aalborg University, Aalborg.

The contribution of article 2 is that it creates an overview of existing commercial rostering systems' support of the created heuristics and characterises the systems, based on both Rostering Approach and level of supported heuristics. The process of evaluating the systems was based on the nine stages of Content Analysis (Kim & Kuljis, 2010).

126 unique systems were retrieved from two software comparison websites (Capterra Inc., ; Nubera, 2016), followed by a screening using inclusion criteria, which reduced the total amount of systems to 77. Of the 77 systems, access was granted to 46 systems, which is the basis of the evaluation. To evaluate the system a set of heuristics was created, based on relevant parameters for IT-supported rostering (Iversen, Kappers, & Pilgaard, 2016) using Affinity Diagramming (Beyer & Holtzblatt, 1997). The focus of the heuristics was similar to Johannessen & Hornbæk's (Johannessen & Hornbæk, 2014) focus on utility, but with a narrower focus on the functionality aspect. To evaluate the systems a Coding Scheme, containing a Codebook and a Coding Form, was used (Kim & Kuljis, 2010). The guidelines provided by Nielsen (Nielsen, 1994) was followed, prescribing multiple evaluators only to confer after their individual evaluations were conducted, in order to eliminate bias. The individual results of the evaluations were compared, if any discrepancies were experienced, the evaluators reached a mutual agreement, in order to get a single data entry of each system. Data was thereafter analysed using Descriptive Statistics (Lynch, 2013).

The result of the analysis consisted of three parts: 1) An overview describing how systems in general supports the created heuristics, 2) Characteristics of systems based on their support of rostering approach, and 3) Characteristics of the systems based on general support level of the created heuristics. The overview presents general tendencies within the evaluated commercial systems and how well each heuristic is supported, based on observations from the evaluations. The second part of the result, groups systems based on their support of Rostering Approach and presents individual characteristics for each group and what they have in common. The third and last part of the results present characteristics of systems based on three mutually exclusive clusters, based on the level of supported heuristics.

2.4 CONTRIBUTION 3

Iversen, S. S., Kappers, D. G. and Pilgaard, C. S. (2016). IT-supported Self-rostering in a Complex Rostering Situation. Department of Computer Science, Aalborg University, Aalborg.

The contribution of article 3 is an evaluation of three proof-of-concept prototypes for IT-supported Self-rostering in complex rostering situations. The process was designed using the ISO standard for Human-Centred Design for interactive systems (ISO/TC 159, 2010), and was focused on designing IT-systems supporting Self-rostering for a specific case. The case is considered complex as it consists of rostering for a residential accommodation for people with mental illnesses, with staffing 24/7, 365 days a year. The residential accommodation employs 46 full time employees plus additional temporaries, interns, and part time employees. Besides the size of the roster, there is a large amount of rules to uphold; organisational rules, laws, and individual agreements. Semi-structured interviews following Kvale's guidelines (Kvale & Brinkmann, 2009) were conducted, in order to understand and specify the context of use by interviewing the current planner of the case, and a planner of a similar rostering situation of high complexity. Effective Prototyping (Arnowitz, Arent, & Berger, 2007) has been incorporated into three activities of the Human-Centred Design approach; Specify the user requirements, Produce design solutions to meet user requirements, and Evaluate the design against user requirements. Three iterations of Human-Centred Design and Effective Prototyping was conducted, with the evaluation of the third iteration being a focus group of employees from the aforementioned case. The focus group was conducted in accordance to the guidelines presented by Greenbaum (Greenbaum, 1998) and Kvale (Kvale & Brinkmann, 2009). The three prototypes that were evaluated were created in the process of Effective Prototyping. The prototypes represented three different approaches for IT-supported Self-rostering.

The results of the focus group were ambiguous in whether or not IT-support of Self-rostering would make sense in complex rostering situations. However, different possibilities, issues and challenges were highlighted, in relation to IT-support of Self-rostering for the given rostering situation. Overall there was scepticism towards the ability for an IT-system to handle the different rules within their rostering situation and giving sufficient feedback on constraint satisfaction. However, if the system would be able to handle constraints and feedback properly, employees saw a mix of the prototypes as potentially implementable. The organisational context was considered important for an IT-system to be feasible, as social dynamics would play a major role in some of the prototype designs. Concerns of imbalanced rosters was raised when the participants were presented to prototypes relying on the employees to create a roster by themselves, as it was reasoned that some employees might only focus on their own needs and wishes. Furthermore, our work raises the question of how IT-supported Self-rostering can be defined, as there is no currently available definition throughout literature.

3 RESEARCH METHODS

In chapter 3, an overview of each article's research method is presented. Along with the research methods, the research question, the research purpose and the outcome from the article is presented. After the overview, the three different articles' research methods are presented in more detail, discussing: strengths, weaknesses and countermeasures.

3.1 OVERVIEW

An overview of the three articles with their respective research method, research purpose, and outcome, have been illustrated in Table 2.

Research Question	**Research Method**	**Research Purpose**	**Outcome**
1. *What parameters should be taken into account when creating IT-supported rostering practices, according to literature?*	Literature Review	Understanding	Framework
2. *How does existing commercial rostering IT-systems support rostering?*	Content Analysis	Understanding	Characteristics of systems
3. *Is it possible to create a viable IT-supported Self-rostering system for a complex rostering situation?*	Basic Research	Change / Understanding	Proof-of-concept

Table 2: Research Method, Research Purpose and Outcome, for each Research Question.

The research method for the first research question is defined as a Literature Review, as defined by Cooper (Cooper, 1985). The research method for answering the second research question is Content Analysis as defined by (Kim & Kuljis, 2010). The research method used to answer the third research question is Basic Research, as defined by Wynekoop & Conger (Wynekoop & Conger, 1990).

Furthermore, the purpose of the research method and outcome of the article has been described for each research question. Each research question will be presented more in depth in the following sections.

3.2 RESEARCH QUESTION 1 - LITERATURE REVIEW

The first research question focus on parameters for IT-supported rostering according to literature. Coopers Taxonomy was used to create the scope of the Literature Review (Cooper, 1985). A challenge for the Literature Review was to find and understand parameters, spread across different research areas, each containing a vast amount of literature. To capture the variety and range of different research areas, the Berrypicking method (Bates, 1989) was used. An illustration of the process of the entire Literature Review can be seen on Figure 1.

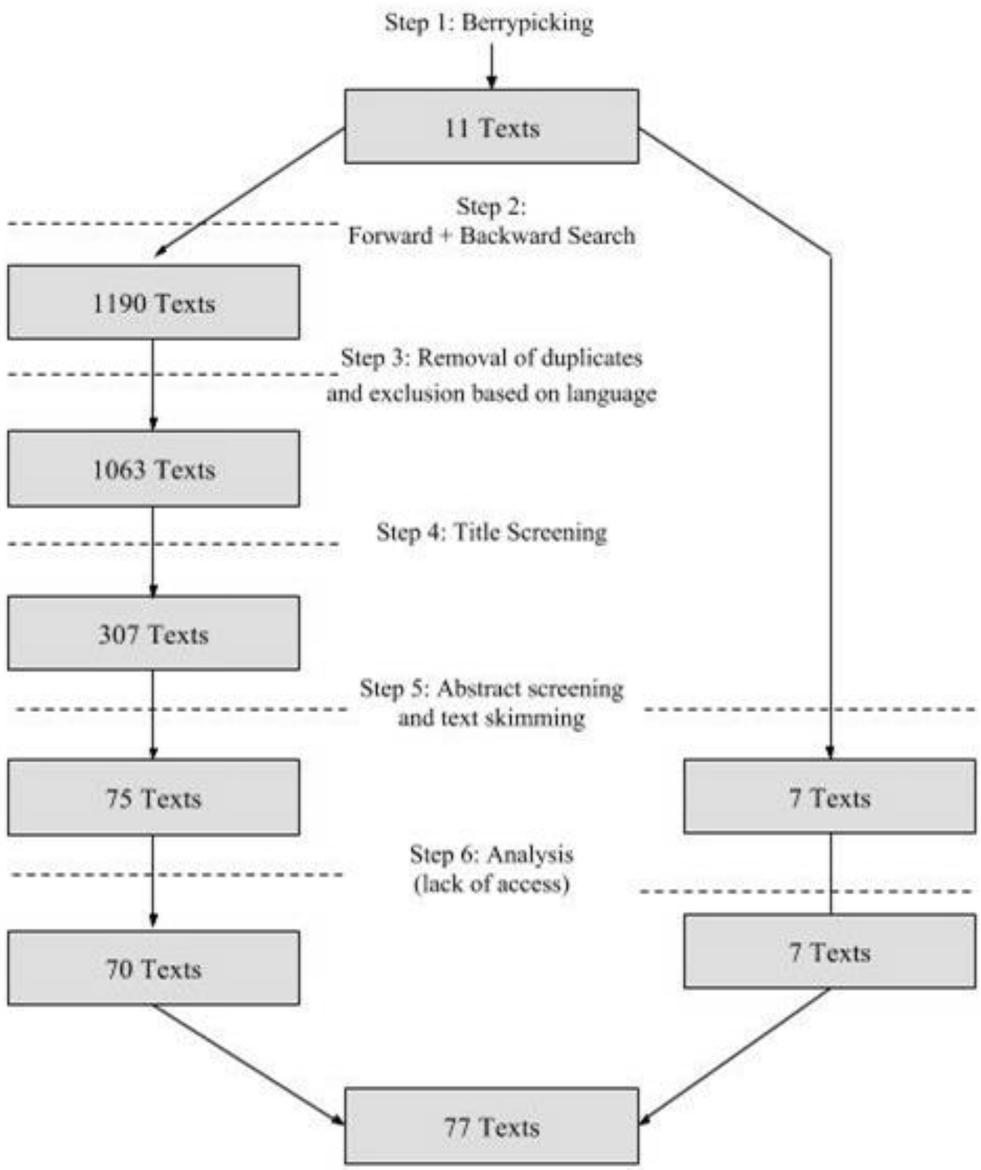

Figure 1: Process of Literature Review process.

The strengths, weaknesses and countermeasures of the research method for article 1 can be seen in Table 3.

Strengths	Weaknesses	Countermeasures
Manage large amount of literature systematically. Identification and selection of literature over multiple areas	Non-systematic Berrypicking of 11 texts.	Inclusion requirements for the 11 texts.
	Representativeness of selected 11 text	Inclusion criteria focusing on multiple areas and perspective.
	Difficult to ensure representativeness.	Forward and backward search. Validation through similar work.
	Not fully exhaustive	No countermeasure

Table 3: Overview of Strengths, Weaknesses and Countermeasures for the Research Question of contribution 1.

The overall strength of using the literature Review research method, is the systematic collection and analysis of literature. By using Berrypicking (Bates, 1989), it was possible to identify and select literature despite the challenges of multiple research areas and a vast amount of literature, using relatively few resources compared to a more systematic approach.

A weakness of our research method was that it was not systematic in the identification and selection of the initial 11 texts. To reduce the impact of the issue, inclusion requirements for the 11 texts are documented along with documentation of the following work to enable replicability.

As the 11 texts were selected from a large amount of literature, the texts may not represent different views on the subject and therefore not be representative of the area. To countermeasure this issue, the inclusion criteria was focused on covering different perspectives and areas of research.

If the previous countermeasure failed, the Citation Searching and Footnote Chasing of the 11 texts could yield non-representative texts and thereby results. As a countermeasure to this, our framework was compared to another established rostering framework.

The Literature Review is not fully exhaustive as it is based only on a portion of the available literature. No countermeasure were taken for this.

3.3 RESEARCH QUESTION 2 - CONTENT ANALYSIS

The second research question focus on existing commercial software systems that supports rostering. The IT-systems were identified and selected through two software comparison websites. A list of heuristics based on relevant parameters for IT-supported rostering was created to evaluate the systems. The heuristics were created based on a perspective on utility and usefulness inspired by Johannessen & Hornbæk (Johannessen & Hornbæk, 2014) from the perspective of potential clients. In order to systematically categorise the systems, a Coding Scheme (Kim & Kuljis, 2010) was created based on relevant parameters for IT-supported rostering (Iversen et al., 2016). The process of Content Analysis is illustrated on Figure 2.

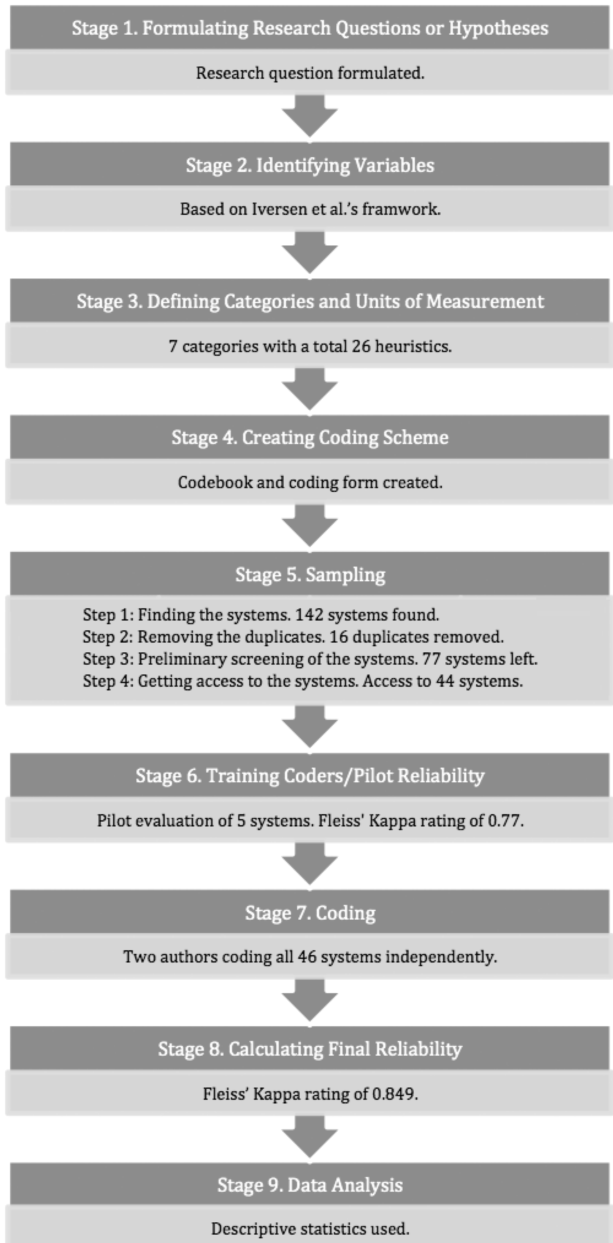

Figure 2: Overview of the Content Analysis process.

The strengths, weaknesses and countermeasures of the research method for article 2 can be seen in Table 4.

Strengths	Weaknesses	Countermeasures
Easy and low cost. Systematic and quantifiable categorisation of utility.	Risk of bias in creation of heuristics and analysis.	Heuristics creation and analysis by multiple authors.
	Heuristics requirements may be interpreted differently.	Pilot test.
	Heuristics based on access to the system.	No countermeasure
	Context insensitive.	No countermeasure
	Static view of dynamic systems.	No countermeasure

Table 4: Overview of Strengths, Weaknesses and Countermeasures for the Research Question of contribution 2.

The strengths of the research method are that it is an easy and low cost way to categorise a large amount of systems, compared to a context centric approach. The Coding Schema (Kim & Kuljis, 2010) was created with an emphasis on the utility aspect of systems. Furthermore, the Coding Scheme allowed for a systematic gathering of quantifiable data, which could be used to categorise systems.

A weakness was that the heuristics for categorisation may be interpreted differently by evaluators and that the creation of the heuristics may be biased itself. To countermeasure this, two researchers were included in creation of the heuristics and evaluation of systems. Furthermore, heuristics were adjusted according to shortcomings or discrepancies in the interpretation of heuristics, based on a pilot test of 5 systems. Incongruence in the final individual evaluations would be compared and mutual agreement would be reached based on providing evidence of heuristic requirements in the systems.

Another weakness was that the systems were tested from a client perspective. The heuristics were therefore composed, based on what is visible and available, as we did not have access to source code. No countermeasures were taken.

Furthermore, the research method is context insensitive, as we tested the systems disregarding the organisational context. No countermeasures were taken.

Another weakness of the research method is that it is a static view of one specific point in time. This is an issue, as the researched systems can change over time due to most of them being web-based. No countermeasures were taken.

3.4 RESEARCH QUESTION 3 - BASIC RESEARCH

The third research question focus on a proof-of-concept prototype of IT-supported Self-rostering in a complex rostering situations. The ISO standard Human-Centred Design for interactive systems (ISO/TC 159, 2010) was used to design and evaluate the proof-of-concept. To Understand and specify the context of use, two semi-structured interviews were conducted following Kvale's guidelines for interviews (Kvale & Brinkmann, 2009). To Specify the user requirements, Produce design solutions to meet user requirements, and finally Evaluate the designs against requirements, three iterations of Effective Prototyping (Arnowitz et al., 2007) were conducted, with the third iteration's evaluation being a focus group of employees from the case. The process of Human-Centred Design can be seen on Figure 3.

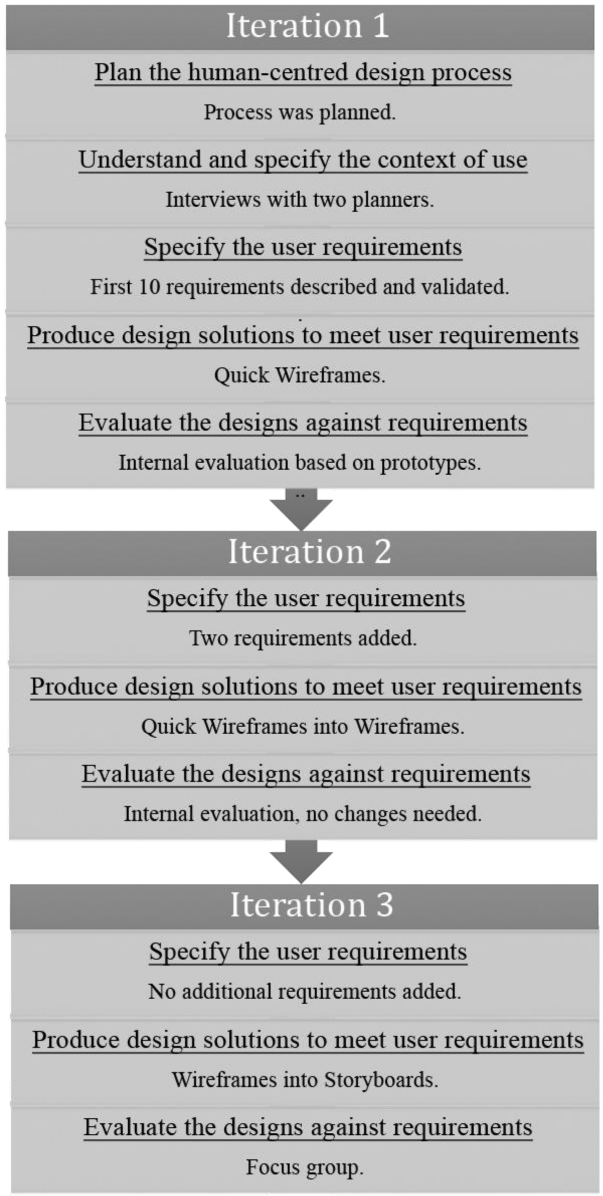

Figure 3: Overview process based Human-Centred Design and Effective Prototyping.

The strengths, weaknesses and countermeasures of the research method for article 3 can be seen in Table 5.

Strengths	Weaknesses	Countermeasures
Exploration of different solutions.	Risk of researcher bias in designs of prototypes and user involvement.	Methods were used to guide processes involving users.
Prototypes are relatable, as they are designed for the employees' context.	Results only applicable to the specific case.	Focus on documentation of method, to improve replicability.
Insights from different stakeholders.	Resource demanding to conduct Proof-of-Concept, because of small strays from focus.	Limit the process according to resources. Timeboxing.

Table 5: Overview of Strengths, Weaknesses and Countermeasures for the Research Question of contribution 3.

The strengths of creating Basic Research in the form of a proof-of-concept is to explore different solutions for a specific context of use. Designing for a specific context allows for more relatable designs and specific implementations. Having used Human-Centred Design, we have interviewed planners to include the organisational perspective of the requirements of an organisation with a complex rostering situation. Furthermore, an evaluation with potential users of the prototype, have been conducted.

When doing exploratory research including users, there is a risk of bias in how one; talk with users, interpret the users, and design the proof-of-concept. To countermeasure this, we used different methods to guide us in different stages of the process. Kvale's guidelines (Kvale & Brinkmann, 2009) was used for interviews, Greenbaum and Kvale (Greenbaum, 1998; Kvale & Brinkmann, 2009) to support the facilitation of the focus group, and Effective Prototyping (Arnowitz et al., 2007) to guide the design of prototypes.

A weakness of proof-of-concepts is the generalisation of results, as the study focus on one specific context. To countermeasure this, we documented our process to make it as replicable as possible.

A weakness of Basic Research is that it is resource demanding compared to other approaches. To countermeasure this, we did two things. Firstly we adapted the process of Human-Centred Design to include Effective Prototyping, in order to limit the process according to resources. Secondly we used Timeboxing (Jalote, Palit, & Kurien, 2004) to manage our time and other resources.

4 CONCLUSION

In this chapter we conclude on each research question and the problem statement. Afterwards, limitations and future work is presented for each contribution.

4.1 RESEARCH QUESTION 1

The first research question was:

- *What parameters should be taken into account when creating IT-supported rostering practices, according to literature?*

To answer this question an interdisciplinary Literature Review was conducted. The Literature Review was based on the initial 11 texts and the 1190 texts found from these. The total of 1201 texts were reduced to 77 through a screening, consisting of 4 steps. The 77 text created the foundation in which relevant parameters for IT-supported rostering was found. The relevant parameters for understanding IT-supported rostering was found to be both complex and contextual in nature. With the purpose of making the knowledge accessible to practitioners, the identified parameters were condensed into a framework. The framework's purpose is to be used as a tool by practitioners when creating or changing IT-supported rostering practices, but also to make it easier to understand the complex and contextual nature of IT-supported rostering. The framework consists of three categories: Strategy, Context, and Implementation. Each Category is defined by an overall purpose and consists of underlying parameters with elaborate examples to better capture the contextual elements in relation to real world problems. The frameworks different categories and parameters are interconnected and should be understood in relation to each other, with no specific starting point.

4.2 RESEARCH QUESTION 2

The second research question was:

- *How does existing commercial rostering IT-systems support rostering?*

Research question 2 was answered by conducting a Content Analysis of existing commercial rostering IT-systems. The Content Analysis was conducted based on 142 systems selected from two software comparison websites, which were reduced to 46 systems by removing duplicates, setting inclusion criteria for system focus, and requesting access. To analyse the 46 systems, a Coding Scheme consisting of 26 heuristics, was created and used independently by two authors to evaluate the systems. Based on the data generated by this evaluation, three different groups of results are presented: 1) An overview describing how systems in general supports the created heuristics, 2) Characteristics of systems based on their support of rostering approach, and 3) Characteristics of the systems based on general support level of the created heuristics.

4.3 RESEARCH QUESTION 3

The third research question was:

- *Is it possible to create a viable IT-supported Self-rostering system for a complex rostering situation?*

To answer this research question, three proof-of-concept prototypes focusing on different approaches for an IT-supported Self-rostering system in a complex rostering situation was designed and evaluated with potential users. The prototypes were designed through a process that combined the ISO standard of Human-Centred Design for interactive systems with Effective Prototyping. In this process two planners were interviewed; one about the specific case in which prototypes would be evaluated, the other about the complications of rostering for a complex situation and which implications a Self-rostering approach could have. This knowledge was used to design three prototypes, which were evaluated through a focus group. The results of the focus group were not definitive, as the presented prototypes was not implemented into the organisation. The focus group was therefore about validation of different ideas of and approaches to IT-supported Self-rostering in a complex rostering situation. Strong tendencies were identified towards a mix of the prototypes being a viable which overall could be advantageous an implementation, if proper preliminary work was conducted and the implementation was robust to accommodate the complexity of rules within the organisation.

4.4 Problem statement

Conclusions from the three research questions are used to answer the problem statement. The problem statement was:

- *How can existing IT-supported rostering systems be analysed using parameters found through literature, and how can this contribute to the creation of new IT-supported rostering systems?*

To analyse existing commercially available IT-systems for rostering, an understanding of relevant parameters was required. This understanding was gained through an interdisciplinary Literature Review, resulting in a framework presenting relevant parameters for IT-supported rostering situations. While the focus of the framework was practitioners and a broad perspective including organisational context, the parameters of the framework was used to create the heuristics which systems were evaluated upon. Data was analysed using descriptive statistics, which showed different tendencies among systems, based on their support of Rostering Approach and level of support for heuristics.

The knowledge gained through the literature and analysis of systems was used to understand how new areas within IT-supported rostering could be explored. The chosen area was IT-supported Self-rostering for complex rostering situations, as Self-rostering through the literature was seen as advantageous, but only advised for less complex rostering situations and did not include the subject of IT-support. Therefore, three prototypes for IT-supported Self-rostering were created through a Human-Centred Design approach, as a proof-of-concept for a case with a complex rostering situation. An evaluation with potential users did not yield definitive conclusion, but showed strong tendencies towards IT-supported Self-rostering being viable for a complex rostering situation.

4.5 Limitations

The limitation for each contribution will be presented in the three following sections.

4.5.1 Contribution 1

The limitations of contribution 1 is primarily tied to the research method. The use of Berrypicking to select texts, excludes the possibility of having an exhaustive Literature Review. Furthermore, the use of Berrypicking can entail a bias, as the initial texts of the Literature Review are selected. While exhaustiveness is unobtainable due to the method, the representativeness can be discussed. Although the Literature Review includes other Literature Reviews and that the product in the form of the framework is validated against similar work, there is still a small risk of not being representative.

4.5.2 Contribution 2

A limitation of the second contribution is that the systems have been evaluated based on self-made heuristics. As we have created the heuristics ourselves, we have no insurance for them being valid or relevant for capturing characteristics for rostering IT-systems. Furthermore, the heuristics relies on what can be seen in the system, some heuristics may therefore not transfer the exact theoretical meaning of the parameters they are based on. Another limitation is that the systems is evaluated by experts, only based on the heuristics, outside of a context.

4.5.3 Contribution 3

A limitation of contribution three is that it is a proof-of-concept based on a single case. The results are therefore limited to the case, which the prototypes are designed for, and tested in. The overall scope and focus on rostering process means that the prototypes have not been implemented. As the prototypes have not been implemented, the tendencies found are partly based on assumptions of how the systems would be when implemented. Lastly, when including users there is a risk of research pleasing, as users might answer what they think researchers wants to hear.

4.6 FUTURE WORK

The following three sections presents future work of the three contributions.

4.6.1 Contribution 1

As we present a framework for a practitioner based on an interdisciplinary Literature Review, aiming to include all relevant parameters for an IT-supported rostering situation, it would be natural for future research to focus on the use of it and test its relevance. By validating and testing the relevance of the framework and its parameters, concern of representativeness could be investigated.

4.6.2 Contribution 2

In contribution two, we evaluate systems based on self-made heuristics and create an overview, and characterise systems based on their support of Rostering Approach and level of support for heuristics. As the heuristics are created by ourselves and the evaluation is conducted by experts, validation of the heuristics and the characteristics of system groups could be areas to focus on. This could be conducted using a more context centric approach to see how findings differ.

4.6.3 Contribution 3

Additional work on testing the viability of the three prototypes would be eminent, as the current evaluation cannot conclude whether or not it is actually possible to implement an IT-supported rostering system into the given context. This would validate the system further, and help towards giving an unambiguous conclusion to the research question. Additional future work could be to implement the same system into contexts similar in size and complexity.

Additionally, it was evident as the prototypes were evaluated, that the three prototypes had implemented Self-rostering in different ways. The different implementations were reasoned to be based on a discrepancy of the understanding of IT-supported Self-rostering. Therefore, it would be relevant to conduct a study on how Self-rostering can be defined in relation to IT-support.

5 BIBLIOGRAPHY

Arnowitz, J., Arent, M., & Berger, N. (2007). Plan your prototype. In D. Simpson, & A. Palmeiro (Eds.), *Effective prototyping for software makers (interactive technologies)* (1st ed.,). San Francisco, CA: Elsevier Inc.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review, 13*(5), 407-424. doi:10.1108/eb024320

Beyer, H., & Holtzblatt, K. (1997). *Contextual design: Defining customer-centered systems* Elsevier.

Burke, E. K., De Causmaecker, P., Petrovic, S., & Berghe, G. V. (2001). Fitness evaluation for nurse scheduling problems. Paper presented at the *Evolutionary Computation, 2001. Proceedings of the 2001 Congress On, , 2.* pp. 1139-1146.

Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling, 7*(6), 441-499. doi:10.1023/B:JOSH.0000046076.75950.0b"

Capterra Inc. *Top employee scheduling software products.* Retrieved 4/25, 2016, from http://www.capterra.com/employee-scheduling-software/

Cheang, B., Li, H., Lim, A., & Rodrigues, B. (2003). Nurse rostering problems—a bibliographic survey. *European Journal of Operational Research, 151*(3), 447-460.

Cooper, H. M. (1985). A taxonomy of literature reviews. Paper presented at the *Annual Meeting of the American Educational Research Association,* Washington, DC. *, 69.* (N/A) pp. 1-49. Retrieved from http://eric.ed.gov/?id=ED254541

Corbin, J. M., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology, 13*(1), 3-21. doi:10.1007/BF00988593

Greenbaum, T. L. (1998). *The handbook for focus group research* (2nd ed.) Sage Publications Inc.

Higgins, A. J., & Postma, S. (2004). Australian sugar mills optimise siding rosters to increase profitability. *Annals of Operations Research, 128*(1-4), 235-249.

ISO/TC 159. (2010). Ergonomics of human-system interaction -- part 210: Human-centred design for interactive systems.

Iversen, S. S., Kappers, D. G., & Pilgaard, C. S. (2016). Parameters to consider when IT-supporting rostering: An interdisciplinary literature review.

Jalote, P., Palit, A., & Kurien, P. (2004). The timeboxing process model for iterative software development. *Advances in Computers, 62,* 67-103.

Johannessen, G. H. J., & Hornbæk, K. (2014). Must evaluation methods be about usability? devising and assessing the utility inspection method. *Behaviour & Information Technology, 33*(2), 195-206. doi:10.1080/0144929X.2012.751708

Kim, I., & Kuljis, J. (2010). Applying content analysis to web based content. Paper presented at the *Information Technology Interfaces (ITI), 2010 32nd International Conference On,* pp. 283-288.

Kvale, S., & Brinkmann, S. (2009). *Interviews: Learning the craft of qualitative research interviewing* Sage.

Leavitt, H. J., & March, J. G. (1962). *Applied organizational change in industry: Structural, technological and humanistic approaches* Carnegie Institute of Technology, Graduate School of Industrial Administration Pittsburgh, Pennsylvania.

Lynch, S. M. (2013). Summarizing data with descriptive statistics. *Using statistics in social research* (pp. 37-20). New York: Springer.

Mason, A. J., Ryan, D. M., & Panton, D. M. (1998). Integrated simulation, heuristic and optimisation approaches to staff scheduling. *Operations Research, 46*(2), 161-175.

Nielsen, J. (1994). Heuristic evaluation. *Usability Inspection Methods, 17*(1), 25-62.

Nubera. (2016). *Employee scheduling software.* Retrieved 4/25, 2016, from https://www.getapp.com/hr-employee-management-software/employee-scheduling/#q=employee&ac=category

Silvestro, R., & Silvestro, C. (2008). Towards a model of strategic roster planning and control: An empirical study of nurse rostering practices in the UK national health service. *Health Services Management Research, 21*(2), 93-105. doi:10.1258/hsmr.2008.007025 [doi]

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2), xiii-xxiii. Retrieved from JSTOR database. Retrieved from http://www.jstor.org/stable/4132319

Wynekoop, J. L., & Conger, S. A. (1990). *A review of computer aided software engineering research methods*. Copenhagen, Denmark.: Proceedings of the IFIP TC8 WG 8.2 Working Conference on The Information Systems Research Arena of The 90's.

Appendix 1

ARTICLE 1: PARAMETERS TO CONSIDER WHEN IT-SUPPORTING ROSTERING: A LITERATURE INTERDISCIPLINARY REVIEW

Parameters to Consider When IT-supporting Rostering: An Interdisciplinary Literature Review

Simon Sønderskov Iversen
Aalborg University
Department of Computer Science
sivers11@student.aau.dk

Daniel Ginnerup Kappers
Aalborg University
Department of Computer Science
dkappe11@student.aau.dk

Christian Sand Pilgaard
Aalborg University
Department of Computer Science
cpilga10@student.aau.dk

ABSTRACT

Rostering is a vast area that is covered by multiple research areas. Despite this, only few have conducted interdisciplinary research on IT-supported rostering, none of which covered all aspects of the subject. In order to give a full overview of the subject, an interdisciplinary Literature Review was conducted. The Berrypicking method was used to select texts which were then screened though multiple steps before getting analysed using Open Coding followed by a categorisation through a process inspired by Affinity Diagramming. This resulted in a total of 13 parameters relevant for IT-rostering. These were categorised into Strategy, Context and Implementation, which helps to capture the contextual complexity involved in rostering theory.

Author Keywords

IT-support; Rostering; Scheduling; Plan; Framework; Parameters; Literature review.

INTRODUCTION

In spring 2016 Google Scholar returned more than 200.000 results when the word roster was searched upon, which shows just how vast the subject of rostering is, as rostering is only a single relevant term out of many. The subject covers several areas of research, despite this, only few have conducted interdisciplinary research on IT-supported rostering, which leads to very few texts giving an overview of parameters relevant to the subject. Of the few texts giving an overview of the subject, none of them included all relations in Leavitt's Diamond [1]; therefore, it was decided to conduct a study, where all relations are taken into account, in order to categorise existing literature according to what they cover. The purpose of this is to give practitioners a thorough description of the parameters relevant in a given rostering situation. This is sought to be accomplished through a framework based on data from an interdisciplinary Literature Review, focusing on IT-supported rostering.

RELATED WORK

In order to get an overview of existing Literature Reviews, they have been evaluated in relation to Leavitt's model for organisational change [1] to show which relations each article covers.

As illustrated in Figure 1, many Literature Reviews cover between one and three relations in Leavitt's model.

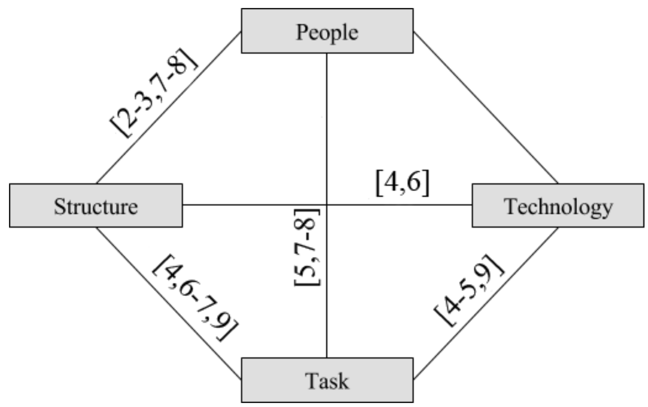

Figure 1. Instantiation of Leavitt's Diamond.

Hayes et al. [2] covers the relation between Structure & People through an article focusing on factors contributing to nurse satisfaction in an acute hospital setting.

Wagstaff & Lie [3] covers the relation of Structure and People, through an article focusing on safety implications within shift work.

Bergh et al. [4] covers the relations of Technology & Structure, Task & Structure, and Technology & Task. This is done through an article focusing on organisational factors relevant for rostering, and how these can be classified into technical features.

Burke et al. [5] cover the relations of Technology & Task, and Task & People. They present an overview of the rostering issue in general including its implications on employees, where they are relating it to technical solution methods.

Ernst et al. [6] cover the relations of Structure & Technology, and Structure & Task. This is done through an article, describing different organisational factors relevant for rostering, and how different technical implementations of rostering IT-system have taken these factors into account.

Moseley et al. [7] cover the relations of People & Task, Structure & Task, and People & Structure. This is done through an article, describing the relevance and importance of retaining experienced nurses from the perspective of an organisation.

Fitzpatrick et al. [8] cover the relations of People & Tasks, and People & Structure. This is done through an article focusing on the organisational impacts, when having different health foci during rostering.

Ibarra-Rojas et al. [9] cover the relations of Technology & Tasks, and Structure & Tasks. This is done through an article, describing a range of different challenges in rostering, such as meeting staffing demands and creating an economically optimised roster.

None of the Literature Reviews however, cover all relations between all four components of the diamond, which would be valuable, in order to understand different parameters relevant for IT-supported rostering, in relation to each other.

The importance of covering all relations is found in the mutual dependency caused by the relations between components [1]. The mutual dependency entails that if a change is made in one component, it will have an effect on the remaining three as well. If the remaining components are not changed accordingly, what is described as a variation, will occur. A variation can decrease the system's capability to accomplish its task, and ultimately invalidate the entire system [2].

SCOPE OF THE LITERATURE REVIEW

Cooper's Taxonomy of Literature Reviews [10] was used as a method to scope the Literature Review. To visualise the scope, an illustration of Coopers Taxonomy, adapted from Vom Brocke et al. [11] was used. The adaptation of the model can be seen on Figure 2.

Characteristic	Categories			
Focus	Research Outcome	Research Methods	Theories	Applications
Goal	Integration	Criticism	Central Issues	
Organisation	Historical	Conceptual	Methodological	
Perspective	Neutral Representation		Espousal of Position	
Audience	Specialised Scholars	General Scholars	Practitioners/ Politicians	General Public
Coverage	Exhaustive	Exhaustive and Selective	Representative	Central/Pivotal

Figure 2. Colored adaption of Coopers Taxonomy, by Vom Brocke.

The focus of attention was determined based on a desire to investigate IT-rostering parameters. A focus on theories was chosen with integration as a goal. The goal was reflected through linguistic bridge-building [10], relating large bodies of literature to broad concepts, and creating a link between these and practices described in case-studies. The literature was organised conceptually, based on neutral representation with practitioners in mind as the audience. Coverage of the literature is representative; as it is based on carefully picked literature intended to cover different aspects of rostering.

SELECTION OF PUBLICATIONS

Initially, the classic model for information retrieval by Stephen E. Robertson [12] was used. A series of search queries were developed in an attempt to cover the subject of rostering and all of its nuances. It was discovered that the amount of literature within the area was vast and hard to separate through keywords and search queries, since the different research areas use different terminology. A query covering the entire subject of rostering, while at the same time returning a reasonable amount of texts, was unobtainable.

As the classic model for information retrieval [12] was insufficient, it was decided to use the Berrypicking model instead [13]. This was chosen because of the model's support of continuous accumulation of data, and that it allows for decisions to be made, as a response to newly realised knowledge.

PROCEDURE

The Berrypicking model [13] is not based on a single query, but rather an evolving search, which helps the researcher to gain increasingly relevant information through each step.

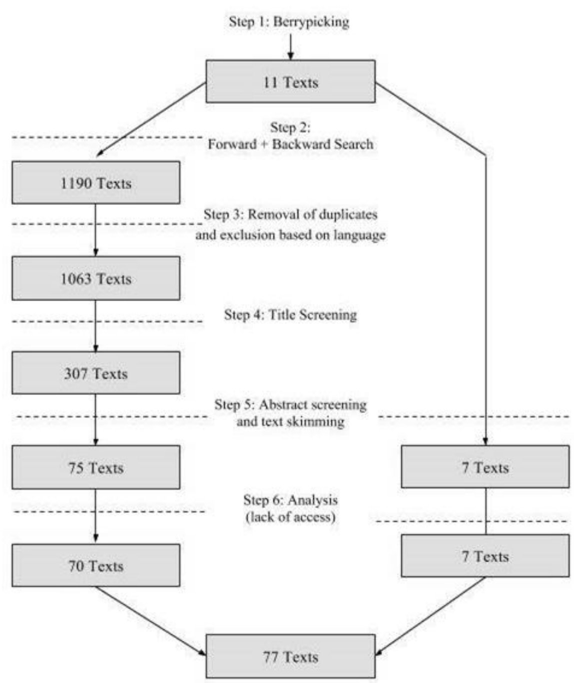

Figure 3. Step-by-Step illustration of screening process.

Schlosser et al. [14] conclude that although evolving search is relevant for research within any field, it is especially relevant for interdisciplinary topics, where terminology has a significant role in collecting research evidence.

Step 1: Selection of Texts Through Berrypicking

It was decided to use the Berrypicking strategy: subject searching [13] to select relevant texts. Information about planning, rostering, scheduling and shift work was accumulated in order to reach a new stage and discover new information. The texts had to be related to rostering, and additionally had to meet at least one of the two following criteria:

1) The text must be multidisciplinary.
2) The text must be comprehensive within its discipline.

11 texts were selected [15-18, L7, L22, L28, L54, L58, L65, L71].

Step 2: Footnote Chasing and Citation Searching

Google Scholar was used to perform Footnote Chasing and Citation Searching [13]; these strategies are also known as Forward and Backward Reference Searching [19]. The searches were performed during fall 2015 and yielded 1190 texts.

Step 3: Removal of Duplicates and Exclusion Based on Language

The 1190 texts were screened, for duplicates and foreign languages. 127 out of the 1190 texts were excluded; 24 were duplicates and 103 were neither in English. This resulted in the 1190 texts being reduced to 1063.

Step 4: Title Screening

A title screening was performed to filter the 1063 texts for relevance. The texts had to fulfil the following criterion:

1) The title of the text must describe a relation to or an aspect of rostering.

The step was performed jointly at plenum, since the criterion was dependent upon subjective assessments. If there were any doubt about relevance, the article would be included, as it would be excluded in the next step if it was found irrelevant. In this step the 1063 texts were reduced to 307 texts. The 11 texts from step 1 were not screened in this step.

Step 5: Abstract Screening and Text Skimming

Because of lack of access, 50 of the 307 texts were excluded as part of the screening. The remaining 257 texts had their abstracts screened, if the criteria were not fulfilled their body text was skimmed. The 11 texts from step 1 were also screened because it was reasoned that they didn't necessarily fulfil the final requirements for step 5. Initially the 11 texts from step 1, and 50 texts from step 4 were screened jointly to achieve a mutual understanding. The remaining texts were screened disjointedly. The text had to fulfil two criteria:

1) The text must be related to either a rostering process or a plan.
2) The text must contain cause and effect from a real context, in the form of results from real life or a simulation based on real life data.

The 11 texts from step 1 were reduced to 7 texts. The 307 texts from step 3 were reduced to 75 texts. This leads to a total of 82 texts.

Step 6 Analysis

Because of lack of full-text access to 5 texts the amount of texts was reduced from 82 to a total of 77 texts for the analysis.

In order to keep track of all the texts while analysing, a table similar to a Concept Matrix [20] was created. While reading the texts, Open Coding [21] was used to develop tags (keywords) that were later used in the analysis. 5 texts were read and tagged jointly at plenum to create a common understanding. As the analysis was being conducted and documented, the tags were combined, and even altered when necessary in accordance to new discoveries.

A framework was constructed using a process inspired by Affinity Diagramming [22]. Each author received pencil and paper, along with the task to illustrate the affinity of the developed tags. The affinities were discussed and titles for each grouping were concluded.

FRAMEWORK FOR IT-SUPPORTED ROSTERING

All references from the Literature Review will be cited with an L prefix. An example of this is: [L1]. The list of references from the Literature Review can be found at: https://goo.gl/tss1H1

Parameters that are relevant when changing IT-supported rostering practices or creating new ones, have been identified. The different parameters' influence on IT-supported rostering have been explained through three categories: Strategy, Context, and Implementation. These categorisations, and corresponding parameters, have been conjoined into a checklist relevant when creating new IT-support rostering practices; the checklist can be seen at Table 1. The parameters will be presented in-depth after the checklist, and should be read and understood in order to properly understand the descriptions in the checklist.

Rostering is complex and has a huge impact not only on the employees or planners, but the entire operation of the organisation. When making changes to the rostering practices there is no single best solution that can be applied in all organisations, since rostering is dynamic and interconnected with the context. The three categories and underlying parameters will be explained individually with a definition of: what they are, why they are important, and how they can influence rostering practices; based on the selected literature. The importance of categories and parameters are not weighted in relation to each other, but the importance can vary depending on context. The order of which parameters are presented is not an expression of a given process, as any parameter can be taken into account at any time. The importance of categories and parameters vary depending on context, and all parameters can be taken into account at any time during the process.

Strategy	
Economy	**Increasing Profit**: Optimise the roster by increasing utility of resources. **Reducing Expenses**: Through construction of the roster, or reducing labour costs associated with the roster.
Employee Satisfaction	**Fairness**: Even distribution of shifts, or an established fairness system. **Motivation**: Increased by employee influence on their own roster, work pace and work goals. **Work-family Balance**: Increased by greater employee influence on the roster and increased flexibility. **Sleepiness**: Reduce by optimal length and placement of shifts, and optimal mix of rotational speed and direction.
Employee Perception	**Homogeneous**: Employees perceived without individual properties besides role/position. **Heterogeneous**: Employees perceived as workers with individual attributes and performance levels.
Context	
Geography	**Planning Location**: Centralisation or Decentralisation of the rostering process. **Distributed Resources**: Utilisation of geographical distribution of resources.
Rules	**Laws**: Regulations set and enforced by the government. **External Rules**: Rules set and enforced in collaboration with external partners. **Policies**: Regulations set and enforced within the organisation.
Work Conditions	**Work Hours**: Length and placement of shifts. **Contracts**: Contractual limitations. **Employee Composition**: Requirements or preferences for employee compositions.
Resources	**Economy**: Resources for development, introduction and maintenance of rostering practices. **Competencies**: Competencies for development, introduction and maintenance of rostering practices.
Implementation	
Rostering Approach	**Self-rostering**: Employees makes the roster themselves. **Team-rostering**: Employees creates partial rosters in groups, which are then combined. **Departmental-rostering**: One appointed planner creates the roster.
IT-Support	**Complete IT-support**: Fully automated rostering system apart from an initial setup. **Partial IT-support**: The range between fully automated and manual rostering. **Analogue rostering**: Manual rostering without IT-support.
Plan Types	**Shifts**: A time-period where an employee has to work. **Assignments**: Pieces of work assigned to employees. **Deadlines**: The latest date and time where a certain activity must be completed.
Rostering Methods	**Pre-planning**: Creation of the first roster made available for employees. **Re-rostering**: Making changes to the published roster.
Procedures	**Rostering Procedures**: Procedures being a part of the rostering process itself. **Organisational Procedures**: Procedures related to rostering indirectly.
Rotation	**Forward Rotation**: Advancing in shift types. **Backward Rotation**: Going backwards in shift types. **Rotational Speed**: The rate at which an employee changes shift type.

Table 1. Checklist of parameters, for practitioners.

Strategy

Strategy is important when changing rostering practices as it helps guide decisions. The identified parameters relevant for the strategy are: Economy, Employee Satisfaction, and Employee Perception.

The two first parameters: Economy and Employee Satisfaction can be perceived as goals for the rostering practice. It is important to understand that the goal for the rostering process can be multi-objective.

Economy

Economy has been identified as optimisable on the following areas: Reduced Expenses and Increased Profit.

Reduced expenses has an extensive focus by the literature relating to economics as a goal. All texts that are categorised as having an economic goal are also focusing on reducing expenses. It can be the main goal, as exemplified by Megeath [L51] where the article is solely describing how to organise and automate the rostering process, minimising the time spent on rostering. It can also be as a sub-goal as exemplified by Dawson et al. [L26] where the welfare of the employees is in focus, but with the thought in mind that healthier employees also minimises the amount of sick-days and accidents on the workplace. There are two ways of reducing the expenses related to rostering: By reducing the expenses of constructing the roster, and by optimising efficiency of the roster, leading to reduced labour costs through the roster as a product.

Most texts focus on the first option, by automating the rostering process using an IT-system; examples of this can be seen in: [L5-L6, L23, L25, L39, L42, L47, L50, L52, L60]. The second option is also represented, common for these articles are that the number of people to roster for, was relatively large [L2, L22, L31, L38, L62].

Increasing Profit is important, as a reduction in expenses, will make the margin of profit greater. An increase in profit can also be accomplished through optimisation of the roster; this is exemplified by Kohl & Karisch [L52] and Yan & Chang [L64], both describing the subject of airline crew rostering. Both articles use the concept of deadhead flights; minimising deadhead flights through planning would reduce expenses and directly affect the margin of profit.

The roster can also be optimised directly in terms of welfare, which can lead to increased profits in terms of decreased sick leave and increased performance. An example is Mason et al. [L6] who describes how a poorly constructed roster can cause a decrease in performance, employee satisfaction and attention level.

Employee Satisfaction

Employee Satisfaction is hard to measure, which is also shown throughout the literature. Employee Satisfaction is divided into four categories: Fairness, Motivation, Work-family Balance and Sleepiness.

Fairness is described differently through the literature, where some describe it as all employees having an equal influence on the roster [L20, L50], while other describe fairness as an even distribution of different shift-types [L49, L61], lastly some are defining fairness as a relation between the organisations needs for coverage, and an employee's need for a customised roster [L39]. Fairness can also be perceived over time; an example is in an article by Gray et al. [L54] who use a predefined point system to determine how fairly every employee is treated.

Motivation is a rather abstract concept, and is among most texts only scarcely described as part of a larger understanding of employee satisfaction. Mason et al. [L6] claim to have created a roster that increases motivation by using a less economically optimised roster. Bassett [L7] discusses motivation in relation to having influence on one's own work. Bassett found that, by letting employees have an influence on their own work pace and work goals, they would be increasingly motivated to perform, and therefore the work performance in general would increase, along with the Employee Satisfaction. This effect on work pace –and goals, would be accommodated through the use of task planning, as the amount of tasks related to time will set a goal to achieve, and a work pace to uphold.

Work-family Balance is rather well-researched and the perfect balance is described as a social construct [L13], where the balancing point is found in a standard day job [L13, L28, L40]. 91.1% of day-worker achieve work-family balance [L40], however only 66% of shift workers achieve this balance. In order to improve work-family balance for shift workers, studies incorporated flexitime [L13, L66], reducing conflict by 28% [L40].

Sleepiness is linked to many other health-related subjects, but is only described in relation to shift work in this Literature Review. It is described by Åkerstedt [L11] that insufficient sleep, or a shift in wake-sleep patterns (e.g. working night shifts) has led to several accidents. The direction and speed of shift rotation is closely related to sleepiness as a consequence. Dawson et al. [L26] describes that the length and placement of shifts in relation each other has a direct effect on the level of fatigue an employee will experience. Sleepiness can also be affected positively by a well-constructed roster. Hesselink et al. [L44] describes how even small changes can affect sleepiness positively. They give an example of how changing the morning shift from a start time of 6AM to 7AM were better fitted to the circadian rhythm and therefore decreased sleepiness remarkably.

Employee Perception

The Homogeneous and Heterogeneous approach to employees are not to be seen as two mutually exclusive options to choose between. It is instead to be perceived as two extremes on a continuous scale [L38].

The Homogeneous extreme of the scale is the situation where employees are seen as resources, and their individual skillset and work performance are not taken into account. Every employee is only valued as the work title they are granted, and are expected to be able to do their job with same speed and efficiency as other with the same title. Examples of this are shown in Curtois' thesis [L23], where the employees are often referred to as a mass, without any distinction between each individual. In an article by Wesson [L77] the same approach is seen, where employees are described as resources.

In the Heterogeneous perception, the employees are seen as individual workers, where attributes and performance of employees are taken into account at an individual level. This is exemplified by Beddoe [L22] who argues that factors such as gender, international status, and personality has an impact on the employee's efficiency in handling tasks, and is therefore taken into account when creating a roster.

Context

Context is the setting for the rostering practice, which sets limits and create possibilities for accomplishing the strategy. The context is by definition not definitive but is hard to change, as it is external influences to the rostering process itself. The identified parameters relevant for the context are: Geography, Rules, Work Conditions, and Resources.

Geography

Geography as a parameter can be explained by splitting it up in two different parts: Planning Location and Distributed Resources.

Planning Location is the location of the planning itself. It is a minor topic in the literature within the Literature Review, where only one article [L58] describe the advantages and disadvantages of centralisation and decentralisation as part of a change. Hill [L58] describes how decentralised timetabling is used, at what they define as a departmental level, in order to allocate rooms between the different courses and departments. A more centralised model with representatives from the departments was chosen as a solution because the problem with inefficiency was rooted in communication disconnects between departments. The change in structure also required change in culture [L58]. Silvestro & Silvestro [L59] describe different types of rostering practices, where Departmental-rostering can be perceived as centralised and Self-rostering is an extreme form of decentralised rostering. Overall, their results indicate that centralisation or decentralisation should be chosen depending on context and the need of the organisation.

Distributed Resources is about optimising the roster in cohesion with geographically distributed organisations or resources. Different locations add to the complexity of rostering and are even more important to take into consideration when it is over great distance, as moving resources becomes increasingly more expensive [L1]. Therefore, when facing geographical challenges one should not only focus on the rostering but also the supporting organisational structure. Abdelghany et al. [L1] describe optimisation of rostering by explaining how they plan for the hub-spoke structure, where most traffic goes through a few locations, which enables them to use standby-crew at these hubs to cover irregularities. This enables them to cover most of the traffic with a relatively inexpensive solution compared to delaying or cancelling flights. Higgins & Postma [L5] describe optimisation of the roster for an Australian sugar mill and its infrastructure. By implementing a system that optimised the roster by increasing utilisation of the transport fleet, they reduced upfront and ongoing labour cost.

Rules

Rules as a parameter is essential to rostering as it represents the views of stakeholders and the viability of a roster. There are different types of rules that relates to rostering depending on the rostering situation. Rules can be classified into three categories: Laws, External Rules and Policies. A specific rule cannot necessarily be placed universally into one category, as what might be a law in some countries may not be in others.

Laws can vary depending on the country and industry, and are something organisations generally have limited possibilities to affect. One example of Laws affecting rostering is the New Zealand airport customs who are required to process 85% of passengers of each arriving flight within one hour through the primary line [L6], which affects how staffing levels of customs are planned. However, Laws that an organisation have to comply with, could also relate to working time, such as the European working time directive that dictates rest periods when working more than 6 hours continuously [L36].

External Rules can come from labour unions or from other forms of partnerships. Labour unions are something that many organisations have to take into account when rostering, as more and more organisations choose to comply with them [L68]. Labour unions are mostly about protecting the employee and ensuring good working conditions [L13], which affect rostering practices as they set some limitations. These limitations can vary depending on the union, sector and country. While some countries by law, enforce maximum continuous and weekly work, in Iceland most of the labour unions have an agreement of 40 hours' workweeks with 8 work hours per day and overtime payment if these are exceeded [L13].

Policies is about enforcing certain behaviour inside an organisation, and consists of internal rule sets, often created to support a strategy. Policies are set and enforced by the organisation itself and are therefore easier to change than Laws and External Rules. Maier-Rothe & Wolfe [L20] gives an example of Policies, as it is hospital policy that nurses have two days off every week, with at least one weekend off, every third week. By enforcing this policy, Maier-Rothe & Wolfe makes the point that it limits the possible number of viable rosters compared to having 4 days off over 2 weeks, which is less strict and therefore more flexible. However, Policies can also relate to procedures and other factors such as distribution of shifts [L49].

Work Conditions

Work Conditions may vary depending on the organisation and can be imposed by Laws, External Rules or Policies. Work Conditions can be split into three categories: Work Hours, Contracts and Employee Composition.

Work Hours affect performance and the well-being of employees and thereby the organisation as a whole. Shift work and night work is proven to reduce performance [L10-L11, L17] and increase fatigue related errors and accidents [L11, L26, L36]. The same consequences are seen when considering increased length of shifts, however not as profoundly. Furthermore, shift work is linked with reduced employee satisfaction, work-family balance and general well-being of employees [L13, L17, L36]. It is not only a matter of having shift work or not, the facilitation and environment of shift work have big impact as well. An example of this is the work of Knauth [L45] who have tested different cyclic and non-cyclic rosters. Based on these tests, Knauth concludes five recommendations for designing shift systems.

Contracts can be an important part of Work Conditions as it set the limitations of both organisational structure and management of employees. Adding flexibility through float staff, part-time staff, or using annualised hours, is something that has to be taken into account in contracts, as these may hinder such approaches. Corominas et al. [L3] works with adding flexibility in the rostering process using annualised working hours without overpay, to accommodate seasonal demand and increase rostering flexibility. Another way to increase flexibility in rostering is to introduce part-time staff. Hur et al. [L24] describes real-time roster adjustment and the amount of part-time employees help to better adjust staffing levels to the demand at a McDonald's. They concluded that higher portion of part-time employees increase flexibility for adjusting employees and demand to increase profit. They also observed that contractual rights of employees limit the possibilities of better adjustments, as management cannot send employees home when they are not needed, without consent. Using part-time employees Mason et al. [L6] was likewise able to increase efficiency of customs staffing at an airport in New Zealand.

Employee Composition can be based on skillset, qualifications, responsibility or preference, and is something many organisations have to consider when rostering. The composition of employees can be affected by Laws, External Rules or Policies, but do not have to be. Rostering for skill mix is prevalent in the literature especially for hospitals where nurses are differentiated depending on their qualifications and responsibility [L22, L32, L39, L46, L56, L61]. Most systems use a hierarchical categorisation of nurses where senior or registered nurses can fill in for a junior or enrolled nurse if needed [L22, L32, L39, L46, L61]. Causmaecker & Berghe [L56] instead use a non-hierarchical approach that allows for more flexibility, but at the cost of a more complex rostering problem. Kohl & Karisch [L52] describe airline crew rostering where employee mixture is based on skills such as language, but also special preference; for married couples to work together on a number of fights over the course of a month, or individuals who cannot work together.

Resources
Resources is about the contextually limiting factors of both Economy and Competencies, when implementing and maintaining new rostering practices.

Implementation of new rostering practices depends on the financial aspects as well as the competencies one has access to. These two aspects are interconnected as financial means can give access to external competencies not present within the organisation. However, Megeath [L51] points out that when one has sophisticated and high quality competencies, one tends to use them even though this may overcomplicate the problem and make a more complicated and expensive solution than necessary.

The time and effort it takes to implement successful organisational change, should be taken into consideration [L68]. The cost of creating and introducing the rostering change is therefore not only in creating the practices but also initial extra cost as a result of the change. Petrovic et al. [L61] created an AI-based rostering system that imitates domain experts' actions and present solutions based on similar previous problems. Such a system will require time to deliver the potential value it is capable of while also requiring adaption from planners. Goodale & Thompson [L38] describe adaptation of individual performance into the rostering process to better accommodate demand and thereby increase profits. They try four different methods where three of them are based on heuristics that can be performed manually if desired. The best heuristic increases profit by 3-4% depending on conditions. The difference in profit depending on approach supports the notion that rostering Competencies in day-to-day rostering can be significant and that IT-systems can help improve rostering.

Implementation
Implementation is about the means of realising the Strategy within the Context. The identified parameters relevant for the implementation are: Rostering Approach, IT-support, Plan Types, Rostering Methods, Procedures and Rotation.

Rostering Approach
Rostering Approach describes different approaches to rostering employees. Three definitions defined by Silvestro & Silvestro [L59] are used: Self-rostering, Team-rostering, and Departmental-rostering.

Self-rostering is defined by the employees making the roster in plenum, and that the employees are responsible for the final roster. Among the benefits of Self-rostering are better employee morale, and increased employee empowerment due to their work pattern preferences being catered for. Their cooperation and teamwork is getting nurtured and they get an understanding of the rostering problems, which leads to greater understanding and more acceptance of compromise. Among the disadvantages are that skill mix may be overlooked as well as over-staffing and under-staffing. Furthermore, no formal procedures may lead to an unbalanced roster overall. Several texts [L22, L47, L61-L62] claim to use Self-rostering, but according to Silvestro & Silvestro's definition [L59], the texts are closer to Team-rostering than Self-rostering. No texts in this Literature Review, beside Silvestro & Silvestro [L59] uses Self-rostering.

Team-rostering is when the roster is made in teams. Silvestro & Silvestro [L59] describe how teams are divided based on competencies. Every team has a leader who is responsible for a partial roster. That leader will meet up with other team leaders to combine their rosters and repair broken constraints if necessary. Among the benefits of Team-rostering are that each team is responsible for rostering their own positions, therefore an optimal skill composition is ensured. The employees are still a part of the rostering process, which leads to an understanding of the rostering process and to better morale. Among the limitations are that conflicts between teams can be difficult to solve, and it can get unmanageable when there are too many teams. There is an administrative burden to combine the rosters and re-work is required if the rosters cannot be put together. Besides Silvestro & Silvestro, the remaining texts covering Team-rostering [L22, L47, L61-L62] are all based on the same case.

Departmental-rostering is the situation where an appointed employee is in charge of the roster and has the responsibility for it being planned. Among the benefits of Departmental-rostering are a balanced rostering design as well as the ability to handle complex situations and still retaining fast decision-making compared to the other Rostering Approaches. Among the limitations are that Departmental-rostering can be perceived as autocratic and the employees can have a lack of understanding for the rostering problems in general, which can lead to employee dissatisfaction and danger of poor morale. The morale can also become low due to danger of favouritism when handling employee requests, which can result in the employees going absent due to their request not being met. Departmental-rostering is by far the most common Rostering Approach in this Literature Review, examples of texts that use this are: [L1-L6, L20, L23-L24, L38-L39, L42, L46, L48-L51, L54, L58, L60, L64]. Some of the texts [L2-L3, L5, L20, L39, L42, L46, L48-L49, L54, L58, L64] do not state that they are using Departmental-rostering but can be interpreted as such, when examining their rostering process. Chun et al. [L4] describes Departmental-rostering as a Rostering Approach that enhances employee utilisation, but does not describe their definition of what Departmental-rostering is.

IT-support

IT-support describes the level of IT-support that is used in the given rostering practice. This range from Complete IT-support to Analogue Rostering. The consequences of IT-support are varying, depending on the context and implementation. A level of IT-support might be appropriate in one situation but not in another due to different contexts.

Complete IT-support is a term that is not defined in the literature. In this Literature Review Complete IT-support is defined as a system that can create rosters without interaction from a user. Within this definition however, it is allowed for the user to make an initial setup, defining constraints and structure, for the system to use as the boundaries for the roster. Whether or not the roster is edited after the creation does not affect the level of IT-support in this case. A large number of the technical texts focus on automating the rostering process, such as: [L2, L20, L32, L37, L42, L46]. An example is Caprara et al. [L2] who describe an automated rostering process that is used to roster employees in a railway setting.

Partial IT-support is a term that covers everything between a Complete IT-support and Analogue Rostering. It is considered partially IT-supported if a system helps a user to roster or if a user helps a system to roster. An example of partial IT-support is the INTERDIP system created by Abdennadher & Schlenker [L60], which is an interactive nurse rostering system that helps creating rosters through interaction with a user.

Analogue Rostering, also known as Self-scheduling in some technical articles [L31], however not to be confused with the Rostering Approach Self-rostering; is what most articles in this Literature Review strive to get away from. Almost all articles in this Literature Review tries to go from an Analogue Rostering process to a more automated process, where more of the rostering is conducted by a system. An example is an article from 1978 by Megeath [L51], where it was attempted to optimise an analogue rostering process by using Mathematical programming, but realised that an analogue rostering process with heuristics were far easier and cheaper to implement, while giving a satisfying result.

Plan Types

Plan Types define how an employee's time is structured. An employee's time can be structured through Shifts, Assignment, and Deadlines.

Shifts is by far the most used factor among the analysed literature. Shifts are defined as assigning a person to a given time slot. Many texts [L1, L4, L6-L9, L11, L15-L16, L18, L22, L28-L29, L31-L32, L36, L37, L39, L42, L44, L46-L47, L49, L55-L56, L58, L61-L62, L65-L69, L72, L75] use this definition. An example of one is Burke et al. [L32] who define employee rostering problems as of assigning a number of Shifts to personnel over a defined period.

Assignments are defined in two ways. Bassett [L7] defines assignments as a piece of work that an employee can be assigned to within a time slot. Farmer & Seers [L33] defines assignments as activities in which efforts may be invested. The two definitions do not necessarily contradict each other, but Farmer & Seers [L33] definition is not reliant on time, so that the assignment itself determines how long an employee should work.

Deadlines are defined by Farmer & Seers [L33] as an ultimate goal that needs to be achieved within a given time period. As a deadline approaches, employees may increase their work rate as response to the deadline closing in [L33].

Rostering Methods

Rostering Methods describe two different methods to use within a rostering situation: Pre-planning and Re-rostering. These should not be seen as mutually exclusive, but rather as complimentary methods. The time at which Pre-planning occurs impacts the outcome of the rostering process as it is based on the available information present at the time. Therefore, a roster can be subject to change as more information is gained and Re-rostering may occur.

Pre-planning is also known as planning ahead of time. Pre-planning is defined as the first iteration of the roster that gets rolled out. Pre-planning has the advantage that employees can plan their private life around the available roster. The disadvantage of Pre-planning is that it is based on information available at the time of the rostering process and thereby not very flexible, unless combined with Re-rostering.

Re-rostering is defined as a change of an existing roster and can be done in real-time, or at any given point after the original roster is released. The advantage of Re-rostering is the possibility of reacting on acute changes in staffing demands. Re-rostering is depicted in many different forms, one of them being Hur et al. [L24] that study the difference between different factors when using real-time roster adjustments. Another example is Moz & Pato [L48] who presents new computational models for re-rostering employees. It is important to handle expectations of when a roster is subject to change and not, as employees use it for planning their personal life and changes may cause dissatisfaction.

Procedures

Procedures describe a way of progression or action. There are two kinds of Procedures that needs to be accounted for: Rostering Procedures and Organisational Procedures.

Rostering Procedures, also referred to as operational constraints, are constraints and procedures that needs to be accounted for, when creating the roster [L2, L4, L20, L22, L37, L63]. An example of such procedure is described in Maier-Rothe & Wolfe's article [L20], where many constraints and procedures are undertaken to allocate nursing staff. Constraints could be: the minimum staffing requirements, maximum number of days allowed to work in a row, number of days off in a week, while procedures could be for an equal distribution of shifts.

Organisational Procedures are not only relevant to the planner, but also the employee, the roster is being created for. An example of such procedure is described by Smith & Wiggins [L46] who researches a company where general rules and procedures are often overridden in order to accommodate customer demands. These overrides leads to an increase in allocation of overtime and float employees.

Rotation

This Literature Review focus on Rotation in relation to shift work. A rotation is characterised by systematically changing the shift type. The most common shift system consists of morning shifts, evening shifts and night shifts, though they vary in start and end time. Days off are not imposed systematically like shift types. The rotation can be either forward or backwards and the speed of rotation can change [L44]. This parameter includes: Forward Rotation, Backward Rotation and Rotational Speed.

Forward Rotation is characterised by advancing in shift types, this could be from morning shifts to evening shift and then to night shifts. From that point, it repeats itself. Guimarães et al. [L27] states that forward rotation favours a better distribution of spare time. When using a forward rotation, the restitution period following a night shift is organised such that the spare time available is concurrent with the restitution period, meaning that the employees uses their own spare time to rest after a night shift [L13]. Knauth [L45] describes how forward rotation is more natural than backward rotation regarding employees' circadian rhythm, therefore it is easier for the employees to fall asleep and avoid sleep disturbance. Åkerstedt [L10] how to reduce sleepiness in shift work and states that the most optimal roster is a slowly forward rotating roster, but that rapid rotation may be preferable in relation to general health.

Backwards Rotation is characterised by going backwards in shift types, for an example by going from night shifts to evening shifts and then to morning shifts. From that point, it repeats itself. Guimarães et al. [L27] states that backwards rotation implies more spare time for the employee. When using backward rotation, the employees spare time does not coincide with the restitution period, therefore the employee will use company hours to rest. Barton et al. [L18] describes how backward rotation has the greatest negative effect on health, including cholesterol, sleep, immune system, gastrointestinal diseases, and fatigue.

Rotational Speed is defined in more than one way in the literature. Hesselink et al. [L44] describes the speed of rotation as how many consecutive shifts of the same type an employee has. They define slow rotation as three consecutive shifts and fast rotation as two consecutive shifts. Geiger-Brown et al. [L36] defines the speed according to how many different shifts an employee has within one week. A fast rotation is several different shifts within 1 week and a slow rotation can be blocks of shift work lasting several weeks [L36]. Different researchers [L12, L16] use specific terms such as rapid rotation, differently between them, therefore it is important to not only gain knowledge of the term used, but also the researchers' definition of the given term. Despite different definitions, Åkerstedt & Wright [L12] warns that when considering rotations, several night shifts in succession will accumulate sleepiness as well as risk of accidents made by the employee.

DISCUSSION

In this section, we will discuss Ernst et al. framework, Leavitt's Diamond, the contextual implication on parameters, diversity of rostering problems, and lastly whether the parameters are limited to an IT-supported context.

Comparison of Frameworks

Throughout this article, a series of parameters in relation to IT-supported rostering have been presented. In order to discuss the comprehensiveness of the parameters, a framework by Ernst et al. [5] have been chosen. The comprehensiveness of the presented framework is tested by checking whether the six modules presented by Ernst et al. are covered by our framework.

Module 1 concerns demand modelling and consists of Task based demand, Flexible demand and Shift based demand, these are covered in Plan Types, Rostering Methods, and Work Conditions.

Module 2 revolves around the determination of days off scheduling. This subject is covered in Rules and Procedures.

Module 3 is about dealing with shift scheduling problems. This part is covered in Plan Types, Rostering Approach and Procedures.

Module 4 concerns line of work construction for each employee. This subject is covered largely in Rules, Procedures, Plan Types and Rotation.

Module 5 revolves around task assignment. This part is covered in Rostering Methods, Procedures and Work Conditions.

Module 6 is about assigning the employees to the roster. This subject is covered in Procedures and Rostering Methods.

The difference between the framework presented in this article and Ernst et al.'s framework [6] is that the former is structure driven while the latter is largely process driven. The process driven framework prescribes a certain process and focuses on the rostering process itself, and is thereby able to be more concrete. The structure driven framework instead focus on understanding the rostering situation independently of a prescribed process and focus more on the connection between the organisation and rostering process. The difference between the two can be described through Leavitt's Diamond; as described in the related work, the framework constructed in this article focus on covering all relations between the four elements in the diamond. The framework by Ernst et al, however, only focus on relations related to the Task element, the other relations are described as related to, but not strictly part of, the rostering process and are therefore excluded from the framework [5].

Leavitt's Diamond

One of the main motivations for creating the framework was to include all four elements of Leavitt's Diamond [1] and take the different relationships between them into account. Leavitt's fundamental idea of that no change can be made in isolation, is taken into account, as the framework is constructed on the premise of being dynamic, without a defined starting point, and with decisions of one parameter affecting others. While the framework is presented from an organisational perspective with a focus on Structure, the elements of; People, Technology and Task, are also taken into account through the parameters presented.

Contextual Implications on Parameters

Every parameter presented is valid in itself; however, the context around the rostering-process is very determinant of each parameters validity for a given situation. All parameters are included equally in the framework, but the validity and importance of parameters will vary depending on the context of use.

Diversity of Rostering Problems

Most of the texts identified in the Literature Review represent rostering problems related to either airline crews or nurses, this could pose a bias towards the parameters identified, however, it seems as though the reason for the large quantity of texts related to either airline crew or nurse rostering, are because these present classic, complex rostering situations. This conclusion is complemented by the fact that, even though they are few, there are texts included in the Literature Review that present other businesses, examples of this is an Australian sugar mill [L5] and a university course scheduling problem [L58].

Non-IT-supported Rostering

In this article, the focus has been to find parameters that relate to IT-supported rostering specifically, however every parameter is described in such general terms that they might be applicable to a large range of rostering situations, and it is very likely that most of the parameters, if not all of them, are also applicable to non-IT-supported rostering situations.

CONCLUSION

A framework for changing or creating new IT-supported rostering practice is presented. The rostering framework consists of three categories: Strategy, Context, and Implementation. The categories have been created to capture the contextual nature and complexity of rostering practices. The underlying parameters for the framework was identified through an interdisciplinary Literature Review.

The limitation for this article is mainly the difficulty in ensuring representativeness. The 11 initial articles have not been chosen systematically, apart from the inclusion criteria and no closure, or exhaustiveness has been ensured throughout the study.

Future work consists of three paths. The first path would be to validate the identified parameters through additional empirical research. The second path would be to conduct additional empirical research with the purpose of finding rostering parameters, not covered by existing literature.

REFERENCES

1. Harold J. Leavitt and James G. March. 1962. Applied organizational change in industry: Structural, technological and humanistic approaches Carnegie Institute of Technology, Graduate School of Industrial Administration Pittsburgh, Pennsylvania.

2. Bronwyn Hayes, Ann Bonner and Julie Pryor. (2010). Factors contributing to nurse job satisfaction in the acute hospital setting: A review of recent literature. *Journal of Nursing Management* 18, 7:804-814. doi:10.1111/j.1365-2834.2010.01131.x [doi]

3. Anthony S. Wagstaff and Jenny-Anne Sigstad Lie. (2011). Shift and night work and long working hours--a systematic review of safety implications. *Scandinavian Journal of Work, Environment &*

Health 37, 3:173-185. doi:10.5271/sjweh.3146 [doi]

4. Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester and Liesje De Boeck. (2013). Personnel scheduling: A Literature Review. *European Journal of Operational Research* 226, 3:367-385. doi:http://dx.doi.org/10.1016/j.ejor.2012.11.029
5. Edmund K. Burke, Patrick De Causmaecker, Greet V. Berghe and Hendrik Van Landeghem. (2004). The state of the art of nurse rostering. *Journal of Scheduling* 7, 6:441-499. doi:10.1023/B:JOSH.0000046076.75950.0b"
6. Andreas T. Ernst, Houyuan Jiang, Mohan Krishnamoorthy and David Sier. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research* 153, 1:3-27. doi:http://dx.doi.org/10.1016/S0377-2217(03)00095-X
7. Amanda Moseley, Lesley Jeffers and Jan Paterson. (2008). The retention of the older nursing workforce: A Literature Review exploring factors that influence the retention and turnover of older nurses. *Contemporary Nurse* 30, 1:46-56. doi:10.5172/conu.673.30.1.46
8. Joanne M. Fitzpatrick, Alison E. While and Julia D. Roberts. (1999). Shift work and its impact upon nurse performance: Current knowledge and research issues. *Journal of Advanced Nursing* 29, 1:18-27.
9. Omar J. Ibarra-Rojas, Felipe Delgado, Ricardo Giesen and Juan C. Muñoz. (2015). Planning, operation, and control of bus transport systems: A Literature Review. *Transportation Research Part B: Methodological* 77, 38-75. doi:http://dx.doi.org/10.1016/j.trb.2015.03.002
10. Harris M. Cooper. (1988). Organizing knowledge syntheses: A taxonomy of Literature Reviews. *Knowledge in Society* 1, 1:104-126. doi:10.1007/BF03177550"
11. Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process. Paper presented at the *Ecis, , 9.* pp. 2206-2217.
12. Stephen E. Robertson. (1977). Theories and models in information retrieval. *Journal of Documentation* 33, 2:126-148. doi:10.1108/eb026639
13. Marcia J. Bates. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review* 13, 5:407-424. doi:10.1108/eb024320
14. Ralf W. Schlosser, Oliver Wendt, Suresh Bhavnani and Barbara Nail-Chiwetalu. (2006). Use of information-seeking strategies for developing systematic reviews and engaging in evidence-based practice: The application of traditional and comprehensive pearl growing. A review. *International Journal of Language & Communication Disorders* 41, 5:567-582. doi:10.1080/13682820600742190
15. Nikolaj G. Borchorst, Susanne Bødker and Pär-Ola Zander. 2009. The boundaries of participatory citizenship. *Ecscw 2009* (pp. 1-20) Springer.
16. Clint Heyer. 2009. High-octane work: The oil and gas workplace. *Ecscw 2009* (pp. 363-382) Springer.
17. Michael J. Muller, Jill Freyne, Casey Dugan, David R. Millen and Jennifer Thom-Santelli. 2009. Return on contribution (ROC): A metric for enterprise social software. *Ecscw 2009* (pp. 143-150) Springer.
18. Kjeld Schmidt, Hilda Tellioglu and Ina Wagner. 2009. Asking for the moon or model-based coordination in distributed design Springer.
19. FAU Libraries. *Guide to science information resources: Backward & forward reference searching.* (2016). Retrieved 05/11, 2016, from http://libguides.fau.edu/c.php?g=325509&p=2182112
20. Jane Webster and Richard T. Watson. (2002). Analyzing the past to prepare for the future: Writing a Literature Review. *MIS Quarterly* 26, 2:xiii-xxiii. Retrieved from http://www.jstor.org/stable/4132319
21. Juliet Corbin and Anselm Strauss. 2008. Basics of qualitative research (3rd ed.): Techniques and procedures for developing grounded theory. *Basics of qualitative research (3rd ed.): Techniques and procedures for developing grounded theory* (0th ed.,) SAGE Publications, Inc. doi:http://dx.doi.org/10.4135/9781452230153
22. Hugh Beyer and Karen Holtzblatt. 1997. Contextual design: Defining customer-centered systems. Elsevier.

Appendix 2

LITERATURE REVIEW REFERENCES OF ARTICLE 1

Parameters to Consider When IT-supporting Rostering: An Interdisciplinary Literature Review

Simon Sønderskov Iversen
Aalborg University
Department of Computer Science
sivers11@student.aau.dk

Daniel Ginnerup Kappers
Aalborg University
Department of Computer Science
dkappe11@student.aau.dk

Christian Sand Pilgaard
Aalborg University
Department of Computer Science
cpilga10@student.aau.dk

The following is the full list of literature, used in the literature review for the first article

REFERENCES

[L1] Abdelghany, A., Ekollu, G., Narasimhan, R., & Abdelghany, K. (2004). A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research, 127*(1-4), 309-331.

[L2] Caprara, A., Toth, P., Vigo, D., & Fischetti, M. (1998). Modeling and solving the crew rostering problem. *Operations Research, 46*(6), 820-830.

[L3] Corominas, A., Lusa, A., & Pastor, R. (2004). Planning annualised hours with a finite set of weekly working hours and joint holidays. *Annals of Operations Research, 128*(1-4), 217-233.

[L4] Chun, A. H. W., Chan, S. H. C., Lam, G. P. S., Tsang, F. M. F., Wong, J., & Yeung, D. W. M. (2000). Nurse rostering at the hospital authority of hong kong. Paper presented at the *Proceedings of the National Conference on Artificial Intelligence,* pp. 951-956.

[L5] Higgins, A. J., & Postma, S. (2004). Australian sugar mills optimise siding rosters to increase profitability. *Annals of Operations Research, 128*(1-4), 235-249.

[L6] Mason, A. J., Ryan, D. M., & Panton, D. M. (1998). Integrated simulation, heuristic and optimisation approaches to staff scheduling. *Operations Research, 46*(2), 161-175.

[L7] Bassett, G. A. (1979). A study of the effects of task goal and schedule choice on work performance. *Organizational Behavior and Human Performance, 24*(2), 202-227.

[L8] Åhsberg, E., Kecklund, G., Åkerstedt, T., & Gamberale, F. (2000). Shiftwork and different dimensions of fatigue. *International Journal of Industrial Ergonomics, 26*(4), 457-465.

[L9] Akerstedt, T. (2003). Shift work and disturbed sleep/wakefulness. *Occupational Medicine (Oxford, England), 53*(2), 89-94.

[L10] Åkerstedt, T. (1988). Sleepiness as a consequence of shift work. *Sleep: Journal of Sleep Research & Sleep Medicine,*

[L11] Akerstedt, T. (1995). Work hours and sleepiness. *Neurophysiologie Clinique/Clinical Neurophysiology, 25*(6), 367-375.

[L12] Åkerstedt, T., & Wright, K. P. (2009). Sleep loss and fatigue in shift work and shift work disorder. *Sleep Medicine Clinics, 4*(2), 257-271.

[L13] Albertsen, K., Grimsmo, Karen Albertsen Kaisa Kauppinen Asbjön, Rafnsdóttir, Björg Aase Sørensen Gudbjörg Linda, & Tómasson, K. (2008). *Working time arrangements and social consequences: What do we know?* Nordic Council of Ministers.

[L14] Angersbach, D., Knauth, P., Loskant, H., Karvonen, M., Undeutsch, K., & Rutenfranz, J. (1980). A retrospective cohort study comparing complaints and diseases in day and shift workers. *International Archives of Occupational and Environmental Health, 45*(2), 127-140.

[L15] Anoop, V. (2011). Shift work: Health, safety and social effects a case of rapidly rotating schedules among male industrial workers.

[L16] Asaoka, S., Aritake, S., Komada, Y., Ozaki, A., Odagiri, Y., Inoue, S., et al. (2013). Factors associated with shift work disorder in nurses working with rapid-rotation schedules in japan: The nurses' sleep health project. *Chronobiology International, 30*(4), 628-636.

[L17] Axelsson, J., Åkerstedt, T., Kecklund, G., & Lowden, A. (2004). Tolerance to shift work—how does it relate to sleep and wakefulness? *International Archives of Occupational and Environmental Health, 77*(2), 121-129.

[L18] Barton, J., Folkard, S., Smith, L., & Poole, C. J. (1994). Effects on health of a change from a delaying to an advancing shift system. *Occupational and Environmental Medicine, 51*(11), 749-755.

[L19] Burke, E., Landa-Silva, D., & Mohamed, B. (2007). On the multi-objective nature of nurse scheduling.

[L20] Maier-Rothe, C., & Wolfe, H. B. (1973). Cyclical scheduling and allocation of nursing staff. *Socio-Economic Planning Sciences, 7*(5), 471-487.

[L21] Caruso, C. C. (2014). Negative impacts of shiftwork and long work hours. *Rehabilitation Nursing, 39*(1), 16-25.

[L22] Beddoe, G. R. (2004). Case-based reasoning in personnel rostering.

[L23] Curtois, T. (2008). Novel heuristic and metaheuristic approaches to the automated scheduling of healthcare personnel.

[L24] Hur, D., Mabert, V. A., & Bretthauer, K. M. (2004). Real-time schedule adjustment decisions: A case study. *Omega, 32*(5), 333-344.

[L25] Warner, D. M., & Prawda, J. (1972). A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science, 19*(4-part-1), 411-422.

[L26] Dawson, D., Noy, Y. I., Härmä, M., Åkerstedt, T., & Belenky, G. (2011). Modelling fatigue and the use of fatigue models in work settings. *Accident Analysis & Prevention, 43*(2), 549-564.

[L27] de M Guimaraes, L. B., Pessa, S. L., & Biguelini, C. (2012). Evaluation of the impact of shiftwork and chronotype on the workers of the imprint and cutting/welding sectors of a flexible packaging manufacturer. *Work (Reading, Mass.), 41 Suppl 1*, 1691-1698. doi:10.3233/WOR-2012-0371-1691 [Ldoi]

[L28] Van Amelsvoort, L. G., Jansen, N. W., Swaen, G. M., Van Den Brandt, Piet A, & Kant, I. (2004). Direction of shift rotation among three-shift workers in relation to psychological health and work-family conflict. *Scandinavian Journal of Work, Environment & Health, ,* 149-156.

[L29] Driscoll, T. R., Grunstein, R. R., & Rogers, N. L. (2007). A systematic review of the neurobehavioural and physiological effects of shiftwork systems. *Sleep Medicine Reviews, 11*(3), 179-194.

[L30] Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research, 140*(2), 266-280.

[L31] Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling, 7*(6), 441-499.

[L32] Burke, E. K., De Causmaecker, P., Petrovic, S., & Berghe, G. V. (2001). Fitness evaluation for nurse scheduling problems. Paper presented at the *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, , 2.* pp. 1139-1146.

[L33] Farmer, S., & Seers, A. (2004). Time enough to work employee motivation and entrainment in the workplace. *Time & Society, 13*(2-3), 265-284.

[L34] Fossum, I. N., Bjorvatn, B., Waage, S., & Pallesen, S. (2013). Effects of shift and night work in the offshore petroleum industry: A systematic review. *Industrial Health, 51*(5), 530-544. doi:DN/JST.JSTAGE/indhealth/2013-0054 [Lpii]

[L35] Vanden Berghe, G. (2002). An advanced model and novel meta-heuristic solution methods to personnel scheduling in healthcare.

[L36] Geiger-Brown, J. M., Lee, C. J., & Trinkoff, A. M. (2012). The role of work schedules in occupational health and safety. *Handbook of occupational health and wellness* (pp. 297-322) Springer.

[L37] Bailey, J. (1985). Integrated days off and shift personnel scheduling. *Computers & Industrial Engineering, 9*(4), 395-404.

[L38] Goodale, J. C., & Thompson, G. M. (2004). A comparison of heuristics for assigning individual employees to labor tour schedules. *Annals of Operations Research, 128*(1-4), 47-63.

[L39] Bard, J. F., & Purnomo, H. W. (2005). Preference scheduling for nurses using column generation. *European Journal of Operational Research, 164*(2), 510-534.

[L40] Jansen, N. W., Mohren, D. C., van Amelsvoort, L. G., Janssen, N., & Kant, I. (2010). Changes in working time arrangements over time as a consequence of work-family conflict. *Chronobiology International, 27*(5), 1045-1061.

[L41] Jung, H., & Lee, B. (2015). Contributors to shift work tolerance in south korean nurses working rotating shift. *Applied Nursing Research, 28*(2), 150-155.

[L42] Weil, G., Heus, K., Francois, P., & Poujade, M. (1995). Constraint programming for nurse scheduling. *Engineering in Medicine and Biology Magazine, IEEE, 14*(4), 417-422.

[L43] Kim, J. W. (2015). Effect of shift work on nocturia. *Urology,*

[L44] Klein Hesselink, J., De Leede, J., & Goudswaard, A. (2010). Effects of the new fast forward rotating five-shift roster at a dutch steel company. *Ergonomics, 53*(6), 727-738.

[L45] Knauth, P. (1996). Designing better shift systems. *Applied Ergonomics, 27*(1), 39-44.

[L46] Smith, L. D., & Wiggins, A. (1977). A computer-based nurse scheduling system. *Computers & Operations Research, 4*(3), 195-212.

[L47] Le, K. N. (2011). A study of evolutionary multiobjective algorithms and their application to knapsack and nurse scheduling problems.

[L48] Moz, M., & Pato, M. V. (2004). Solving the problem of rerostering nurse schedules with hard constraints: New multicommodity flow models. *Annals of Operations Research*, *128*(1-4), 179-197.

[L49] Sodhi, M. S., & Norris, S. (2004). A flexible, fast, and optimal modeling approach applied to crew rostering at london underground. *Annals of Operations Research, 127*(1-4), 259-281.

[L50] Warner, D. M. (1976). Scheduling nursing personnel according to nursing preference: A mathematical programming approach. *Operations Research, 24*(5), 842-856.

[L51] Megeath, J. D. (1978). Successful hospital personnel scheduling. *Interfaces, 8*(2), 55-60.

[L52] Kohl, N., & Karisch, S. E. (2004). Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research, 127*(1-4), 223-257.

[L53] Niu, S. F., Chu, H., Chung, M. H., Lin, C. C., Chang, Y. S., & Chou, K. R. (2013). Sleep quality in nurses: A randomized clinical trial of day and night shift workers. *Biological Research for Nursing, 15*(3), 273-279. doi:10.1177/1099800412439459 [Ldoi]

[L54] Gray, J., McIntire, D., & Doller, H. (1990). Nurses' preferences for specific work schedules. Paper presented at the *Proceedings of the Annual Symposium on Computer Application in Medical Care,* pp. 885.

[L55] Ohayon, M. M., Lemoine, P., Arnaud-Briant, V., & Dreyfus, M. (2002). Prevalence and consequences of sleep disorders in a shift worker population. *Journal of Psychosomatic Research, 53*(1), 577-583.

[L56] De Causmaecker, P., & Berghe, G. V. (2003). Relaxation of coverage constraints in hospital personnel rostering. *Practice and theory of automated timetabling IV* (pp. 129-147) Springer.

[L57] Palhares, V. d. C., Corrente, J. E., & Matsubara, B. B. (2014). Association between sleep quality and quality of life in nursing professionals working rotating shifts. *Revista De Saude Publica, 48*(4), 594-601.

[L58] Hill, D. L. (2008). Qualitative timetabling: An organizational and qualitative approach to improving university course scheduling. *College Quarterly, 11*(3), n3.

[L59] Silvestro, R., & Silvestro, C. (2000). An evaluation of nurse rostering practices in the national health service. *Journal of Advanced Nursing, 32*(3), 525-535.

[L60] Abdennadher, S., & Schlenker, H. (1999). INTERDIP–an interactive constraint based nurse scheduler. Paper presented at the *Proceedings of the First International Conference and Exhibition on the Practical Application of Constraint Technologies and Logic Programming, PACLP,*

[L61] Petrovic, S., Beddoe, G., & Berghe, G. V. (2002). Case-based reasoning in employee rostering: Learning repair strategies from domain experts.

[L62] Petrovic, S., Beddoe, G., & Berghe, G. V. (2003). Storing and adapting repair experiences in employee rostering. *Practice and theory of automated timetabling IV* (pp. 148-165) Springer.

[L63] Topaloglu, S., & Ozkarahan, I. (2004). An implicit goal programming model for the tour scheduling problem considering the employee work preferences. *Annals of Operations Research, 128*(1-4), 135-158.

[L64] Yan, S., & Chang, J. (2002). Airline cockpit crew scheduling. *European Journal of Operational Research, 136*(3), 501-511.

[L65] Drake, C. L., Roehrs, T., Richardson, G., Walsh, J. K., & Roth, T. (2004). Shift work sleep disorder: Prevalence and consequences beyond that of symptomatic day workers. *Sleep, 27*(8), 1453-1462.

[L66] Šimunić, A., & Gregov, L. (2012). Conflict between work and family roles and satisfaction among nurses in different shift systems in croatia: A questionnaire survey. *Archives of Industrial Hygiene and Toxicology, 63*(2), 189-197.

[L67] Smith, M. R., & Eastman, C. I. (2012). Shift work: Health, performance and safety problems, traditional countermeasures, and innovative management strategies to reduce circadian misalignment. *Nature and Science of Sleep, 4*, 111.

[L68] Sun, N., & Maniatis, T. (2014). Scheduling in the context of resident duty hour reform. *BMC Medical Education, 14*(Suppl 1), S18.

[L69] Takahashi, M., Nakata, A., Haratani, T., Otsuka, Y., Kaida, K., & Fukasawa, K. (2006). Psychosocial work characteristics predicting daytime sleepiness in day and shift workers. *Chronobiology International, 23*(6), 1409-1422.

[L70] Thompson, G. M. (2005). Using information on unconstrained student demand to improve university course schedules. *Journal of Operations Management, 23*(2), 197-208.

[L71] Schümmer, T., Tellioglu, H., & Haake, J. M. (2009). Towards living Agendas—Shaping the next generation of business meetings. *Ecscw 2009* (pp. 263-282) Springer.

[L72] Tucker, P., Smith, L., Macdonald, I., & Folkard, S. (2000). Effects of direction of rotation in continuous and discontinuous 8 hour shift systems. *Occupational and Environmental Medicine, 57*(10), 678-684.

[L73] Vangelova, K. K. (2008). Stress and fatigue in sound engineers: The effect of broadcasting in a life show and shift work. *Central European Journal of Public Health, 16*(2), 87.

[L74] Vangelova, K. K. (2008). Variations of cortisol, fatigue and sleep disturbances in sound engineers: Effect of job task and fast backward-rotating shifts. *Reviews on Environmental Health, 23*(1), 83-89.

[L75] Tharmmaphornphilas, W., & Norman, B. A. (2004). A quantitative method for determining proper job rotation intervals. *Annals of Operations Research, 128*(1-4), 251-266.

[L76] Warner, D. M. (1976). Scheduling nursing personnel according to nursing preference: A mathematical programming approach. *Operations Research, 24*(5), 842-856.

[L77] Wesson, D. A. (1995). The interactive effects of rules on teaching timetable flexibility and resource utilization. *Higher Education, 30*(3), 305-312.

Appendix 3

ARTICLE 2: PARAMETERS TO CONSIDER WHEN IT-SUPPORING ROSTERING: A LITERATURE INTERDISCIPLINARY REVIEW

A Review of Existing Commercial Rostering IT-systems

Simon Sønderskov Iversen
Aalborg University
Department of Computer Science
sivers11@student.aau.dk

Daniel Ginnerup Kappers
Aalborg University
Department of Computer Science
dkappe11@student.aau.dk

Christian Sand Pilgaard
Aalborg University
Department of Computer Science
cpilga10@student.aau.dk

ABSTRACT
This article evaluates existing commercial rostering IT-systems. This was done through a Content Analysis, where rostering parameters relevant to IT-supported rostering, found in the literature, were turned into 26 heuristics, and subsequently used to categorise 46 systems. The results have been analysed using Descriptive Statistics and are split into an overview of how each heuristic is supported, a categorisation based on Rostering Approach, and a categorisation based on the amount of heuristics that are supported.

Author Keywords
Rostering; IT-systems, Review; Parameters; Rostering Approach; Self-rostering, Team-Rostering, Departmental-rostering; Automation

INTRODUCTION
Since the early 1970s researchers have modelled rostering situations mathematically [1], in order to create models that could be used to optimise rosters. In the last four and a half decades a leap in technology has taken place. For most people and businesses, the Internet was born in 1995 [2], which made the concept of online systems interesting to software developers. A vast amount of both online and offline rostering systems have been created and many studies have been conducted in regards to implementing or optimising IT-supported rostering [3-4]. Additionally, many academic reviews have been conducted, concerning IT-supported Rostering Approaches and technical techniques [5-6], however none have made a review on how commercial IT-systems support rostering. It was therefore found of interest to make a review on how existing commercial rostering IT-systems support rostering.

RELATED WORK
Related work for this article is a collection of texts, each classifying different IT-systems, as no article that classifies rostering systems have been found. Different terms are used within the related work. To be consistent in presentation of the related work, the terminology has been standardised. We have chosen to use heuristics as a term covering the criteria which systems are classified upon.

Park et al. [7] compares online stores, selected through comparison websites by dividing them into categories. The authors used Content Analysis and constructed a Codebook with heuristics, in order to categorise the systems. The heuristics were chosen through a discussion and validated by two domain-experts. The results were described using Descriptive Statistics and chi-square tests. The results are split into 8 categories, and further divided into 43 heuristics; the systems were related to the heuristics by describing how large a percentage of the systems, fulfilled the criteria.

Norris et al. [8] compares pro-anorexia websites by using a search term suggested through a suggestion-engine tool. The search term was used as input on the top-three search engines, which were found through two comparison websites. 20 websites were included using undisclosed criteria, but the analysis of websites were stopped after analysing 12 websites, as it was decided that enough data was collected. Open coding was used for the analysis, where the tags were constructed through a textual analysis, done individually by three researchers, followed by a statistical analysis so that they were able to report basic Descriptive Statistics from their results. The results are split into 10 heuristics, and the fulfilment described through the percentages of systems fulfilling the given heuristic. Additionally, the websites were split thematically.

Moore & Ayers [9] compares postnatal mental health websites by using the four largest search engines based on their own assessment, and then used four search terms they found through discussion. For each search the first 25 results, and the links found within these, were analysed for inclusion based on undisclosed criteria. 114 websites were included in the study and analysed based on different scales, developed in cooperation with a domain specialist. The authors used Descriptive Statistics to report their findings. The results are split into 6 heuristics, and the systems are related to these through statistics showing how many systems fulfil a given heuristic.

Bingley et al. [10] compares tourism websites that are included based on popularity; the inclusion process is not described further. Subsequently the websites were screened with the inclusion criterion of having user-generated content, resulting in a total of 22 websites. Content Analysis was used and due to lack of suitable classification techniques, the authors developed their own. To explain the results, the authors used Descriptive Statistics. The results were split into two categories, further split into five classifications, and finally split into 15 heuristics. The heuristics were related to the reviewed systems through how large a percentage of systems fulfilled the heuristic, but also by comparing each system, to each heuristic individually.

Common for all of the related work are that they use Descriptive Statistics to explain their results. The Descriptive Statistics are however adapted for each article, in order to communicate their results more strongly. Also common for the articles are that they all study the content of IT-systems.

METHOD

Content Analysis was used as an overall method to follow throughout the study, as it is the most appropriate method for descriptive research, according to Babbie [11]. The process of this study is illustrated in Figure 1. Kim & Kuljis' [12] method for Content Analysis of web-based content is used. The method aids in the analysis of how rostering parameters found through literature, are supported in existing commercial IT-systems for rostering. Their 9-stage process model, which is based on Neuendorf [13], is used with a few changes as described at each stage.

Stage 1. Formulating Research Questions or Hypotheses

This article strives to find out how existing commercial rostering IT-systems support rostering, as it helps to identify characteristics, which were used to group systems. This has led to the following research question: "*How does existing commercial rostering IT-systems support rostering?*".

Stage 2. Identifying Variables

The identification of variables was based on an existing framework for IT-supported rostering [14]. No need for further conceptualisation was needed, as the framework describes the parameters conceptually.

It was decided to scope the selection of parameters to only cover some aspects of the parameters, as some were tightly related to organisational context and others would require a very subjective assessment. The parameters chosen are shown in Table 1.

Stage 3. Defining Categories and Units of Measurement

7 categories were developed using Affinity Diagramming [15], to cover the selected parts of the selected framework [14]. Table 1 presents different rostering parameters along with their factors from Iversen et al.'s framework. The numbers next to the factors corresponds to following 7 categories, that were developed in this stage: (1) Constraints, (2) Automation, (3) Rostering Approach, (4) Procedures, (5) Reporting, (6) Planning Units and (7) Employee. A minus sign (-) has been added next to the parameters that are not covered.

Stage 4. Creating Coding Scheme

The authors developed a Coding Scheme consisting of a Codebook along with a Coding Form, as prescribed by Kim & Kuljis [12]. The Codebook was inspired by some of the

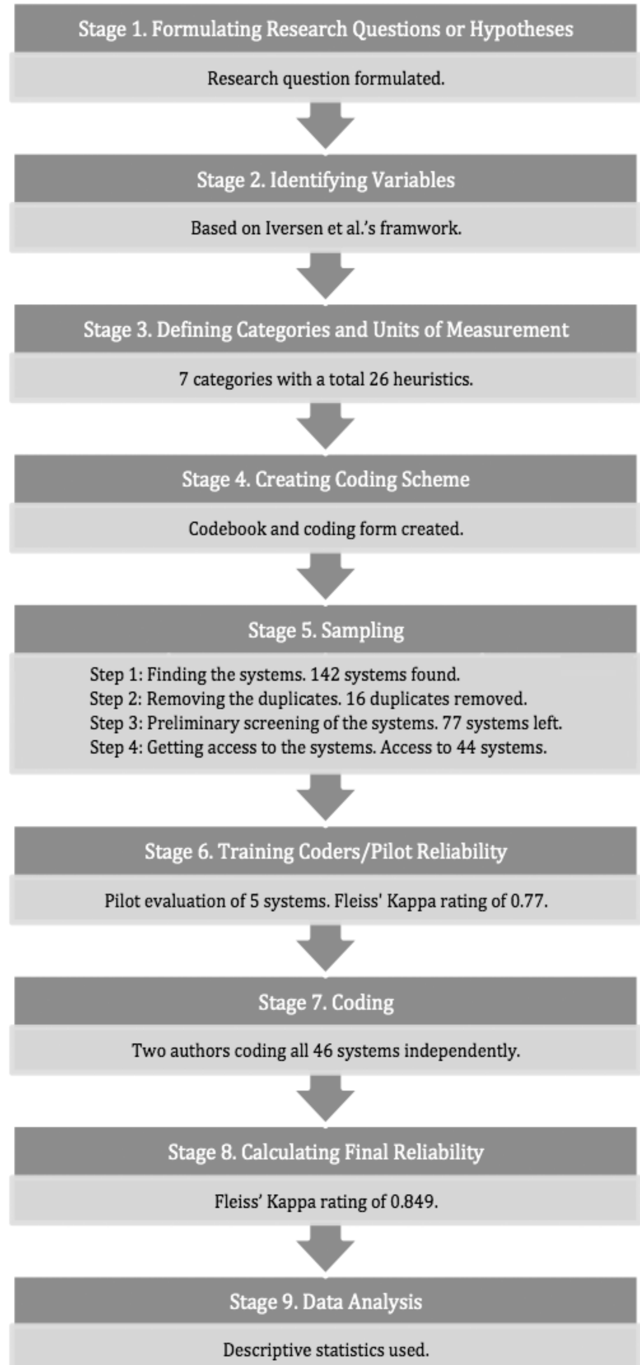

Figure 1. Content Analysis process model, adapted from Kim & Kuljis [12].

related work [16-18]. The variables of the Codebook are referred to as heuristics, as they needed some form of subjective interpretation from the evaluator [19]. The heuristics themselves are focussed on the system's ability to support rostering from a perspective of utility and usefulness [20]. The finalised Codebook can be found in Table 2. In order to evaluate whether a system supported a heuristic, the evaluators had to prove that the heuristic was valid in at least one case. The two categories: Planning

Units and Employee, had special conditions in order for them to be proved. The heuristics in Planning Units had to be explicitly specified in the planning block. The heuristics in Employee had to have an effect on rostering, by either visual clues, warnings or constraints

Stage 5. Sampling

Sampling of the systems consisted of four steps: finding the systems, removing the duplicates, preliminary screening of the systems, and getting access to the systems.

Step 1: Finding the Systems

In spring 2016, the software comparison websites Capterra [21] and Getapp [22] were used to retrieve lists of potentially relevant systems. All systems in the category Employee Scheduling from both sites were included. A list of 142 systems was retrieved, where Capterra provided 79 systems and Getapp provided 63 systems.

Step 2: Removing the Duplicates

A comparison of the 142 systems revealed 16 duplicates. The duplicates were excluded, resulting in a list of 126 unique systems.

Step 3: Preliminary Screening of the Systems

A screening was conducted independently by two authors, based on available online material. If the authors had different results they would confer and resolve the differences. The systems were screened for the following inclusion criteria:

- One of the primary goals of the system has to be, to roster work hours of employees.
- The system has to support different meeting times for individuals.

77 of the 126 systems fulfilled the inclusion criteria.

Step 4: Getting Access to the Systems

Access to the 77 systems were requested either through their website or their public e-mail. All companies were given 3 weeks to respond and grant access to their system. Access to a total of 46 systems were granted within the allotted time. The companies of the remaining systems did either not respond, declined access, or informed that their systems were discontinued.

Stage 6. Training Coders/Pilot Reliability

5 of the 46 systems were randomly selected for a pilot evaluation, conducted independently by two authors. After the pilot evaluation, Inter-coder Reliability was calculated using Fleiss' Kappa. The result revealed a Fleiss' Kappa rating of 0.77, which according to Landis & Koch [23] is equal to a substantial strength of agreement. This is also above the acceptable indicator of 0.75 provided by Banerjee et al. [24]. Following the pilot evaluation, the Codebook was revised to make heuristics more specific, in an attempt to increase the common understanding between the authors and eliminate uncertainties in heuristics.

Rostering parameters	Factors
Strategy	
Economy	Increasing Profit (5) Reducing Expenses (5)
Employee Satisfaction	Fairness (5) Motivation (-) Work-family Balance (-) Sleepiness (-)
Employee Perception	Homogeneous (7) Heterogeneous (7)
Context	
Geography	Planning Location (3) Distributed Resources (6)
Rules	Laws (1) External Rules (1) Policies (1)
Work Conditions	Work Hours (1) Contracts (1) Composition (7)
Resources	Economy (5) Competencies (7)
Implementation	
Rostering Approach	Self-rostering (3) Team-rostering (3) Departmental-rostering (3)
IT-support	Complete IT-support (2) Partial IT-support (2) No IT-support (2)
Plan Types	Shifts (6) Assignments (6) Deadlines (-)
Rostering Methods	Preplanning (4) Re-rostering (4)
Procedures	Rostering Procedures (1) Organisational Procedures (4)
Rotation	Forward Rotation (2) Backward Rotation (2) Rotational Speed (2)

Table 1. Overview of heuristics related to rostering parameters.

Stage 7. Coding

The same two authors who performed the previous step, also coded all 46 systems independently, using the final version of the Codebook, including the 5 systems that were included in the pilot evaluation.

System supports	Possible answers	Criterion for inclusion, is Yes if
Rostering Approach		
Self-rostering	Yes/No	An employee had the possibility to take a shift without the approval of a manager, or if the employee had the final decision whether to work or not.
Team-rostering	Yes/No	The system was designed for partial rosters where team-leaders had access and permission to edit the partial rosters for their team.
Departmental-rostering	Yes/No	The authors could make a finalised roster without acceptance from employees.
Automation		
Generation Of Rosters Without Interference	Yes/No	The system could create a roster for more than one week, taking constraints and all employees into account.
Generation Of Smaller Parts Of Rosters	Yes/No	The system could create or redo smaller parts of the roster, for at least one day and not the whole roster, taking constraints and multiple employees into account.
Ability To Override Part Of A Roster Scheduled By The System	Yes/No/(N/A)*	The authors could make changes in a roster created by the system.
Optimise Current Roster Itself	Yes/No	The system could optimise an existing roster, on at least one parameter, without user intervention.
Make Suggestions For Optimisation	Yes/No	The system could suggest single or multiple optimisations that required the authors' help or approval.
Rotation	Yes/No	The system could implement rotation through automation or if it could force rotation in a roster through constraints.
Cyclic	Yes/No	The system could implement repetition of the roster.
Procedures		
Requests For Work	Yes/No	The employee could request to work a specific timeslot within a given day, including open shifts.
Requests For Leave	Yes/No	The employee could request leave for a specific timeslot, day or period of time.
Trade/give/take Shifts	Yes/No	The employees could trade, give or take shifts, regardless of it having to be approved or not.
Planning Units		
Assignments	Yes/No	The system supported management of assignments or tasks within a shift.
Responsibility	Yes/No	The system supported responsibilities on a shift.
Location	Yes/No	The system supported a specific location for a shift.
Reporting		
Index Of Economy	Yes/No	The system showed the authors a preview of the cost or economic consequences of the roster, or when the authors did an action and were shown an economic consequence thereof.
Index Of Employee Satisfaction	Yes/No	The system showed the authors a preview of individual employee satisfaction, including fairness.
Employee		
Individual Performance Level	Yes/No	Performance level could be set individually for each employee.
Role	Yes/No	A role could be set for each employee.
Extra Qualifications	Yes/No	Extra qualifications could be set for each employee.
Scheduling Influence	Yes/No	Individual preferences could be set for each employee.
Constraints		
Hard constraints	Yes/No	Authors could set constraints in the system or edit existing system constraints, which could not be broken, unless explicitly approved.
Soft constraints	Yes/No	Authors could set constraints in the system or edit existing system constraints, which could be broken without explicit approval.
Multi-Level constraints	Yes/No	The system used a hierarchical or weighted system to prioritise constraints.
Ability to override constraints	Yes/No/Partly/(N/A)**	Authors could override any type of constraints set in the system.

Table 2. Final version of the Codebook.

*N/A was selected if the system did not support any of the two heuristics above. **Partly was selected if authors could override some, but not all constraints set in the system. N/A was selected if the system did not support constraints.

Stage 8. Calculating Final Reliability

9 systems were evaluated each day, where one author started from the top of the list and the other author at the bottom. The authors shared the same login, so data from one author was available to the other author. A rule was made, so that the authors were not allowed to make hints in the systems. After all systems were evaluated, Inter-coder Reliability was calculated using Fleiss' Kappa. The results revealed a Fleiss' Kappa rating of 0.85, which according to Landis & Koch [23] is equal to an almost perfect strength of agreement. At the end of the evaluation, disagreements were settled by letting the author in favour of a requirement prove to the other author that a system fulfilled the requirement. Systems were marked according to mutual agreement.

Stage 9. Data Analysis

Descriptive Statistics was used to analyse the data. The results from the data analysis are described in Results.

RESULTS

The results are divided into three sections: an overview of the different categories with their heuristics, characteristics based on Rostering Approach, and clusters based on level of support.

Overview

In order to get a general understanding of how commercial IT-systems supports rostering, an overview of the heuristics and the number of systems supporting it, have been created. The overview can be found in Table 3. Notable findings of the different categories of heuristics will be presented in this section.

Rostering Approach

Rostering Approach consists of three heuristics: Self-rostering, Team-rostering and Departmental-rostering. The options are not mutually exclusive; one system can therefore be in all three categories if it fulfils the heuristic.

Most distinctive is the high percentage of systems supporting Departmental-rostering at 97.8% (n=45). Only one system [S20] did not support Departmental-rostering, as it was specialised towards event coordinators with varying workforce.

Team-rostering and Self-rostering was supported by 50% (n=23) and 45.7% (n=21) of the systems respectively. While this is lower than Departmental-rostering, it is still a relatively large percentage of the existing rostering IT-systems supporting what might be considered more untraditional Rostering Approaches.

Automation

Automation consists of 7 heuristics: Generation Of Rosters Without Interference, Generation Of Smaller Parts Of Rosters, Ability To Override Part Of A Roster Scheduled By The System, Optimise Current Roster Itself, Make Suggestions For Optimisation, Rotation and Cyclic.

Heuristic	Supported in number of systems (n=46)	Low Support (n=10)	Medium Support (n=21)	High Support (n=15)
Rostering Approach				
Self-rostering	21 (45.7%)	20%	43%	67%
Team-rostering	23 (50%)	10%	48%	80%
Departmental-rostering	45 (97.8%)	90%	100%	100%
Automation				
Generation Of Rosters Without Interference	12 (26.1%)	10%	10%	60%
Generation Of Smaller Parts Of Roster	12 (26.1%)	0%	14%	60%
Ability To Override Part Of A Roster Scheduled By The System	16 (100%)**	100%	100%	100%
Optimise Current Roster Itself	0 (0%)	0%	0%	0%
Make Suggestions For Optimisation	7 (15.2%)	0%	10%	33%
Rotation	1 (2.2%)	0%	5%	0%
Cyclic	41 (89.1%)	60%	95%	100%
Procedures				
Requests For Work	29 (63%)	10%	71%	87%
Requests For Leave	37 (80.4%)	40%	86%	100%
Trade/give/take Shifts	33 (71.7%)	20%	76%	100%
Planning Units				
Assignments	12 (26.1%)	20%	19%	40%
Responsibility	43 (93.5%)	80%	95%	100%
Location	40 (87%)	60%	95%	93%
Reporting				
Index Of Economy	29 (63%)	20%	57%	100%
Index Of Employee Satisfaction	30 (65.2%)	30%	62%	93%
Employee				
Individual Performance Level	5 (10.9%)	0%	10%	20%
Role	43 (93.5%)	80%	100%	93%
Extra Qualifications	10 (21.7%)	0%	24%	33%
Scheduling Influence	34 (73.9%)	40%	76%	93%
Constraints				
Hard Constraints	28 (60.9%)	30%	67%	73%
Soft Constraints	38 (82.6%)	70%	90%	80%
Multi-Level Constraints	2 (4.3%)	0%	5%	7%
Ability To Override Constraints	43 (97.7%)*	80%	100%	93%

Table 3. Overview of constraints and how they are supported in rostering IT-systems.

* Total (n=44), and options Yes and Partly are combined in the overview. ** Total (n=16).

Generation of both the whole roster without interference, and smaller parts of the roster was supported by 26.1% (n=12) of systems. The overlap of systems between the two heuristics is 33.3% (n=4). The heuristic Ability To Override Part Of A Roster Scheduled By The System is supported by 100% (n=16) of systems that incorporate automatic generation of a roster or parts of it.

Another notable finding was that no system included the ability to optimise an existing roster. However, 15.2% (n=7) of systems included the ability to Make Suggestions For Optimisations, but required user interference to implement each suggestion.

Rotation was only supported by 2.2% (n=1) of systems, where the one system supporting it [S11] did so through the use of Constraints that could be set to enforce rotations.

89.1% (n=41) of systems supported Cyclic repetition of the roster, often through the use of template-like functions, applying a predefined roster.

Procedures

Procedures consist of three heuristics: Requests For Work, Requests For Leave and Trade/give/take Shifts.

63% (n=29) of systems supported for an employee to request for work. Requests For Work was often identified by the employee being able to request to take an open shift, which would later be reviewed by a planner.

Requests For Leave was supported by 80.4% (n=37) of systems, which was the highest within the category of Procedures. Requests For Leave was often in the form of employees having the opportunity to submit requests for vacation.

The heuristic concerning employees' ability to trade, give or take shifts widely supported, as 71.7% (n=33) of systems allowed employees to Trade/give/take Shifts.

Planning Units

Planning Units consists of three heuristics: Assignments, Responsibility and Location.

The least supported heuristic was Assignments only being supported by 26.1% of evaluated systems. However, a lot of systems used a general notes field, which by Organisational Procedures could be used as Assignments.

Most systems supported the heuristic of Responsibility tied to a shift, at 93.5% (n=43). The responsibility was often in the form of a role or function, but not all systems had Responsibility tied to employee roles directly.

Location was prevalent among the evaluated systems as 87% (n=40) of systems supported it. The support of Location was often through support of different rosters tied to specific locations, but some systems included locations for specific shifts.

Reporting

The category Reporting consists of two heuristics: Index Of Economy and Index Of Employee Satisfaction.

Index Of Economy was supported by 63% (n=29) of evaluated systems. The aspect of economy was often tied to the salary of employees, which were used to calculate the cost of the roster.

The heuristic of Index Of Employee Satisfaction is supported by 65.2% (n=30) of systems. However, the support of Employee Satisfaction is only evaluated in terms of Fairness and the ability to differentiate individually in terms of request or preferences being fulfilled.

Employees

Employees consists of four heuristics*:* Individual Performance Level, Role, Extra Qualifications and Scheduling Influence.

Individual Performance Level was supported by 10.9% (n=5) of systems. Individual Performance Level was most often represented by a predefined numeric scale in the systems supporting it.

Roles was prevalent, as 93.5% (n=43) of systems included the ability to assign employees roles which were actively used in the rostering. This was often done at a universal level where employees were assigned these roles afterwards.

Constraints

Constraints consists of four heuristics: Hard Constraints, Soft Constraints, Multi-Level Constraints and Ability To Override Constraints.

Systems were evaluated for their support of Hard Constraints, Soft Constraints, and Multi-Level Constraints. Soft Constraints was most prevalent with 82.6% (n=38) of systems supporting it, while Hard Constraints was supported by 60.9% (n=28) and only 4.3% (n=2) of systems supported Multi-Level Constraints. The two systems [S40-S41] supporting Multi-Level Constraints did so by tying constraints to a predefined scale of severity.

Almost all 44 systems supporting constraints, allows the Constraints to be either Fully or Partly overwritten as 97.7% (n=43) allows for it.

Characteristics Based on Rostering Approach

This section describes the characteristics of the evaluated systems, based on their support of Rostering Approach. In order to accomplish this, the systems have been grouped by the three heuristics within the category Rostering Approach. A deviation threshold of 20% was used when examining similarities between the systems within a group. It is important to note that the Partly option of Ability to Override Constraints have been seen as the same as the answer Yes, as it somewhat support ability to override as a system characteristic.

General tendencies

General tendencies cover what the three groups of systems have in common. When comparing the 3 groups they have 7 heuristics in common: Cyclic, Ability to Override Constraints, Role, Location, Responsibility, Requests For Leave and Departmental-rostering. This was not surprising in itself, as each of these heuristics was supported by over 80% of all systems. However, the heuristic Soft Constraints is not included in this list even though it was supported by 82.6% (n=38) of systems.

The seven heuristics can be seen as basic functionality of rostering IT-systems regardless of their support of Rostering Approach.

Self-rostering

The Self-rostering group consists of 21 systems and supports 5 heuristics besides the common heuristics that every group supports. 2 out of the 5 heuristics were unique to only Self-rostering, while the other 3 overlapped with either the Team-rostering or Departmental-rostering group.

Of the two unique heuristics, one was self-explanatory as it was the heuristic of Self-rostering. The other heuristic unique to this group was Scheduling Influence, which was a heuristic focusing on employees being able to set individual preferences in the system. The prevalence of both of these heuristics shows that this group of systems had a greater focus on including the employees as individuals and valued the opportunity of the employees to have influence on the roster.

The three heuristics, supported by this and one other group of systems, were: Soft Constraints, Requests For Work, and Trade/give/take shifts. The heuristics of Requests For Work and Trade/give/take Shifts being supported by this group focus on employee involvement. No logical explanation of the relation with the inclusion of Soft Constraints have been found, besides the fact that it was generally well supported as 82.6% (n=38) of systems supported it.

Team-rostering

The Team-rostering group of systems consists of 23 systems and supports 2 unique heuristics and 2 heuristics overlapping with another group, beside the common heuristics that all three groups support.

The Team-rostering group supported the heuristic of Team-rostering as the only group, which is to be expected. The other unique heuristic to the Team-rostering group of systems was the support of Index Of Employee Satisfaction. It indicates that this group of systems valued employee satisfaction. However, as Index Of Employee Satisfaction was often based on statistics of individual fulfilment of requests, the systems in this group may instead focus on control and a more data driven Rostering Approach, as Team-rostering is based on hierarchal management.

The two heuristics: Requests For Work and Trade/give/take Shifts, overlapped with one other group. The support of these two heuristics shows that employees typically had the ability request for work and trade, give or take shifts. The second heuristic does not prescribe that an action needs approval, but combined with the layer of middle management a higher level of control often follows.

Departmental-rostering

The Departmental-rostering group of systems consists of 45 systems. This group supported 0 unique heuristics and only 1 heuristic which was also supported by one of the other groups, besides the common heuristics supported by all three groups.

The Departmental-rostering group is different than the other two groups, as it consists of 45 systems only leaving out a single system [S20]. The characteristics of this group due to its size are related to the overall support of heuristics. Therefore, the main finding of this group is how all but one system included this approach.

The 1 heuristic overlapping with another group, was Soft Constraints. As the support of Soft Constraints on a general level was 82.6% (n=38), the Departmental-rostering group was expected to include this as well.

Clusters Based on Level of Support

The systems have been split into three clusters, according to the number of parameters supported by each system. Table 4 shows the number of supported parameters for each classification. The numbers of parameters were chosen based upon characteristics in the data.

Number of parameters supported	Classification
4-10	Low Support
11-15	Medium Support
16-21	High Support

Table 4. Classification of systems based on number of parameters supported.

Table 5 shows how each cluster is characterised by the categories, with an average score at the bottom. The value in each cell is an expression of the level of coverage the cluster provides for the given category. The following scale has been used to group the percentage values into a description for the level of support:

- 0-24%: Almost none,
- 25-49%: Limited
- 50-74%: Intermediate
- 75-100%: Fully

Category	Low (%)	Medium (%)	High (%)
Rostering Approach	Limited (40)	Intermediate (64)	Fully (82)
Automation	Almost none (24)	Limited (33)	Intermediate (50)
Procedures	Almost none (23)	Fully (78)	Fully (96)
Planning Units	Intermediate (53)	Intermediate (70)	Fully (78)
Reporting	Limited (25)	Intermediate (60)	Fully (97)
Employees	Limited (30)	Intermediate (53)	Intermediate (60)
Constraints	Limited (45)	Intermediate (66)	Intermediate (63)
Average	Limited (34)	Intermediate (61)	Fully (75)

Table 5. Score based on the average of each category for each cluster.

Low Support

10 systems [S5-S6, S8, S13-S14, S20-S21, S27, S42, S46] have been classified as having a Low Support level of heuristics. The coverage ranged from 4 to 10 heuristics, with an average coverage of 7.6 heuristics. Low Support systems are rated lower in 24 heuristics compared to the average of all systems. The remaining 2 heuristics were equal to the average, where the average results were either 0% or 100%, as seen in Table 3.

Common for systems in this cluster are that 8 systems [S5-S6, S13-S14, S20-S21, S27, S42, S46] only supported one Rostering Approach, which shows that the systems in the cluster were focused on what approach they wanted to support. Only one system [S20] supported Self-rostering exclusively, and was to be found in this cluster. The rest of the systems [S5-S6, S8, S13-S14, S21, S27, S42, S46] supported Departmental-rostering, which shows that this Rostering Approach was prevalent.

When the category of Automation was examined, only one system [S6] supported any kind of automation, besides the heuristic Cyclic, in this cluster. This shows that the systems in this cluster did not cater to rostering situations with high complexity, as the planners did not get any automated help to solve complex rostering problem. 6 of the systems [S5-S6, S8, S21, S27, S46] in this cluster supported Cyclic, which, when examining the other clusters, shows to be an indicator of how this heuristic was basic functionality in many of the evaluated systems.

The systems in this cluster supported Reporting remarkably little, compared to the systems in other clusters. 3 of the systems [S8, S27, S42] supported one kind of reporting, while 1 system [S46] was the only one that supported both types of reporting, which imply that there was little focus on reporting in the systems within this cluster.

When comparing the systems that support Scheduling Influence in this cluster, with other clusters, the systems in this cluster only supported about half as many percentage points as the others. This result reveals how this cluster only supported limited employee involvement.

Hard Constraints were supported little, when comparing the systems in this cluster with the systems from the other clusters. The little support for Hard Constraints shows that systems in this cluster did not force constraints to be upheld. 2 systems [S13, S42] did not support any constraints, as the only two systems of all evaluated systems.

The systems in this cluster can generally be described as specialised systems, with little support for the heuristics in general. The Low Support cluster sets the level of supported heuristics, which the other two clusters only adds to.

Medium Support

21 systems [S1, S3, S11-S12, S15-S16, S18-S19, S22-S25, S28-S30, S34, S36, S39-S40, S44-S45] have been classified as having a Medium Support of heuristics. The coverage ranged from 11 to 15 heuristics, with an average coverage of 13.43 heuristics. When comparing how close the heuristics in this cluster were to the average of all systems, the results are: 14 heuristics were within 5 percent points, 10 heuristics were within 10 percent points and 2 heuristics more than 10 percent points. The two heuristics that showed the highest difference from average were both related to Automation and can be found in Table 3.

7 systems [S1, S11, S19, S25, S34, S39-S40] in this cluster supported one Rostering Approach, while 9 systems [S3, S12, S15, S18, S24, S28-S30, S44] supported two Rostering Approaches and 5 systems [S16, S22-S23, S36, S45] supports three Rostering Approaches. The systems in this cluster had a tendency to support multiple Rostering Approaches, which shows that the systems were more generalised than the systems in the cluster Low Support.

Automation was in general only supported by 6 systems [S11, S24-S25, S29, S36, S39] in this cluster, when examining the systems that supported Cyclic. Cyclic had risen from a support of 60% in the cluster Low Support, to 95% support in this cluster, as seen in Table 3. This can be explained, as Cyclic being basic functionality in the systems.

When comparing how different categories are placed near each other, in accordance with their clusters, different results were revealed. When relating the cluster with Low Support to this one, the support for the categories: Procedures, Planning Units, Employees and Constraints, were the ones that develop the most. It is worth noting that the average level of support for Constraints was higher in this cluster, than the High Support cluster. The systems support of the heuristic Trade/give/take Shifts went from 20% support in the cluster Low Support, to 76% in this cluster, also seen in Table 3. When going from this cluster to the cluster High Support, the support for Automation was the category that developed the most. The level of which, systems support Reporting develops the same, when comparing clusters.

The systems in this cluster can generally be described as having sporadic support for different heuristics. The only common characteristic was that it was not normal for the systems in this cluster to support the category Automation, apart from the heuristic Cyclic.

High Support

15 systems [S2, S4, S7, S9-S10, S17, S26, S31-S33, S35, S37-S38, S41, S43] have been classified as having High Support of heuristics. The coverage ranged from 16 to 21 heuristics, with an average coverage of 17.73 heuristics.

8 systems [S4, S7, S9, S33-S34, S38, S41, S43] supported two Rostering Approaches and 7 systems [S2, S10, S17, S26, S31-S32, S37] supported three Rostering Approaches. The systems in this cluster supported multiple Rostering Approaches, which shows that the systems had a High Support for different Rostering Approaches. This can be explained, as an attempt by the systems within this cluster, to support a high amount of different rostering situations.

When examining Automation, 13 systems [S2, S4, S7, S9-S10, S26, S31-S33, S35, S38, S41, S43] supported some kind of Automation that is not Cyclic. Two systems [S10, S43] only supported one more heuristic in Automation besides Cyclic, while the others support more. When the systems support more heuristics, they are able to support more complex rostering situations. Due to the systems supporting complex rostering situations, the systems try to help the users, by offering Automation that help easing the complex situation.

All systems in this cluster supported all heuristics in the category Procedures, beside two systems [S33, S41] that are the only ones that did not support the heuristic Request for Work. This can be explained, by the fact that the systems in this cluster generally allow the user to do almost anything within the system.

Likewise did the systems in this cluster support all heuristics in the category Reporting, besides from one system [S4], which did not support Index Of Employee Satisfaction. When comparing the category Reporting across the three clusters, it clearly shows that Reporting was increasingly supported by the systems, as the systems' average level of support rose.

The systems in this cluster can in general be described as systems that allow and supported for their users to do almost everything within the system. Categories like Procedures and Reporting were almost always supported, while Automation is sometimes supported. When examining this cluster, every single system, except two [S4, S7], supported the Ability to Override Constraints, where one system [S4] supported it Partly. This supports the notion that the systems in this cluster allowed and supported their users to do almost everything within the system.

DISCUSSION

The work closely related to this article generally uses the same methodology as we do. Differences between the articles in related work, and this article, are mostly practical differences based on the subject of the study. Small differences in the presentation of the results are also present, however they are evidently with the same purpose, all of them describing the heuristics, and then relating the percentage of systems that fulfil the given heuristic.

When reviewing the gathered data, it was discovered that the expectation of very few systems supporting Self-rostering, did not comply with the data. As the heuristic is an attempt to construct a scenario, covering the practical implementation of the theoretical presentation of Self-rostering. It was reasoned that the discrepancy between expectation and reality, was due to the formulation of the heuristic. The intent of the formulation for Self-rostering was clear, however the resulting formulation was discovered as being too broad.

The two comparison sites that were used to retrieve lists of potentially relevant systems, provided a total of 142 systems, where 16 systems (11%) were duplicates, leaving 126 unique systems to be preliminary screened. Due to the little overlap it was decided to compare the systems with a third comparison website after the analysis had taken place. The new comparison website [25] revealed 117 systems, of which 19 (16%) were duplicates when compared to our original list, leaving 98 new systems. Based on the inclusion percentage from the earlier screening, it would be expected that 60 of the 98 systems would be included for further analysis.

The authors were granted access to the systems, on the same level as potential clients of the systems would be. This entails that findings are limited to what the systems presented and how the authors interacted with the systems. The authors was aware of the limitations related to the lack of access to the source code, therefore the Codebook with heuristics was developed with this in mind. Despite this, the level of access still poses an uncertainty, as the authors who analyse have to incite the same results. In order to ensure some stability in the results, inter-coder reliability using Fleiss' Kappa was calculated for the authors' results from the pilot evaluation, as well as the full evaluation.

A point of criticism on our method could be, that the systems have not been analysed in a relevant context of use, which could lead to a misunderstanding of how the systems are meant to be used. However, Heuristic Evaluation prescribe that the frame of reference is the same for all evaluators [19]. It is typical for expert evaluations to be conducted outside of context of use, as seen in related work [7-10].

CONCLUSION

Content Analysis has been used as an overall method for the process within this study. The existing commercial rostering IT-systems were selected through two comparison websites. A Codebook with 7 categories containing 26 heuristics, was based on parameters from Iversen et al.'s framework [14]. The results were analysed using Descriptive Statistics. The result of the analysis consisted of three parts:

- An overview describing how systems in general support the created heuristics.
- Characteristics of systems based on their support of Rostering Approach.
- Characteristics of the systems based on general support level of the created heuristics.

Limitations consists of two areas. The first area is that the authors were granted access to the systems on the same level as potential clients would be, which affected the way the heuristics were developed, and thereby the results. The second area was that the systems only were looked at without a context, and therefore limits what could be measured, since the same system can be utilised in different ways depending on organisational contexts.

Future work consists primarily of two paths. The first path is to validate the results found by testing the heuristics developed in this article by including more systems and replicate the process. The second path is to test and evaluate whether the heuristics from this article are applicable, when examining the systems in different contexts.

REFERENCES

1. D. M. Warner and Juan Prawda. (1972). A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science* 19, 4-part-1:411-422.
2. Manuel Castells. 2002. The internet galaxy: Reflections on the internet, business, and society Oxford University Press on Demand.
3. Joe D. Megeath. (1978). Successful hospital personnel scheduling. Interfaces 8, 2:55-60.
4. Andrew J. Higgins and Steve Postma. (2004). Australian sugar mills optimise siding rosters to increase profitability. Annals of Operations Research 128, 1-4:235-249.
5. Edmund K. Burke, Patrick De Causmaecker, Greet V. Berghe and Hendrik Van Landeghem. (2004). The state of the art of nurse rostering. Journal of Scheduling 7, 6:441-499. doi:10.1023/B:JOSH.0000046076.75950.0b"
6. Timothy Curtois. (2008). Novel heuristic and metaheuristic approaches to the automated scheduling of healthcare personnel.
7. Hyun-Hee Park, Qin Li and Jung-Ok Jeon. (2009). A study of comparison analysis on online visual merchandising practices between korea and china. *Journal of the Korean Society of Clothing and Textiles* 33, 6:916-928.
8. Mark L. Norris, Katherine M. Boydell, Leora Pinhas and Debra K. Katzman. (2006). Ana and the internet: A review of pro-anorexia websites. *International Journal of Eating Disorders* 39, 6:443-447.
9. Donna Moore and Susan Ayers. (2011). A review of postnatal mental health websites: Help for healthcare professionals and patients. *Archives of Women's Mental Health* 14, 6:443-452.
10. Scott Bingley, Stephen Burgess, Carmine Sellitto, Carmen Cox and Jeremy Buultjens. (2010). A classification scheme for analysing web 2.0 tourism websites. *Journal of Electronic Commerce Research* 11, 4:281.
11. Earl R. Babbie. (1973). *Survey Research Methods*
12. Inhwa Kim and Jasna Kuljis. (2010). Applying content analysis to web based content. Paper presented at the *Information Technology Interfaces (ITI), 2010 32nd International Conference on*, pp. 283-288.
13. Kimberly A. Neuendorf. 2002. *The content analysis guidebook*. Sage.
14. Simon S. Iversen, Daniel G. Kappers and Christian S. Pilgaard. (2016). *Parameters to Consider when IT-supporting Rostering: An Interdisciplinary Literature Review*.
15. Hugh Beyer and Karen Holtzblatt. 1997. Contextual design: Defining customer-centered systems. Elsevier.
16. Ji Hye Park and Leslie Stoel. (2002). Apparel shopping on the internet: Information availability on US apparel merchant web sites. *Journal of Fashion Marketing and Management: An International Journal* 6, 2:158-176.
17. Lola Khakimdjanova and Jihye Park. (2005). Online visual merchandising practice of apparel e-merchants. *Journal of Retailing and Consumer Services* 12, 5:307-318.
18. Judy C. Lin and Hsipeng Lu. (2000). Towards an understanding of the behavioural intention to use a web site. *International Journal of Information Management* 20, 3:197-208.
19. Jakob Nielsen. (1994). Heuristic evaluation. *Usability Inspection Methods* 17, 1:25-62.
20. Guðrún H. J. Johannessen and Kasper Hornbæk. (2014). Must evaluation methods be about usability? devising and assessing the utility inspection method. *Behaviour & Information Technology* 33, 2:195-206. doi:10.1080/0144929X.2012.751708

21. Capterra Inc. *Top employee scheduling software products.* Retrieved 4/25, 2016, from http://www.capterra.com/employee-scheduling-software/

22. Nubera. *Employee scheduling software.* (2016). Retrieved 4/25, 2016, from https://www.getapp.com/hr-employee-management-software/employee-scheduling/#q=employee&ac=category

23. J. R. Landis and Gary G. Koch. (1977). The measurement of observer agreement for categorical data. *Biometrics* 159-174.

24. Mousumi Banerjee, Michelle Capozzoli, Laura McSweeney and Debajyoti Sinha. (1999). Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics* 27, 1:3-23.

25. Inc Software Advice. *Compare employee scheduling software.* Retrieved 5/26, 2016, from http://www.softwareadvice.com/hr/employee-scheduling-comparison/

SYSTEMS

S1. Aladtec, http://aladtec.com/

S2. NimbleSchedule, http://nimbleschedule.com/

S3. eSchedule, http://eschedule.ca/

S4. 7shifts, http://7shifts.com/

S5. Ento, http://ento.com/

S6. RotaCloud, http://rotacloud.com/

S7. Staffjoy, https://staffjoy.com/

S8. TimeForge Scheduling, http://timeforge.com/

S9. TrackSmart, http://tracksmart.com/

S10. WhatTimeDoIWork, http://whattimedoiwork.com/

S11. WhenToWork, http://whentowork.com/

S12. FloorSchedule, http://floorschedule.info/

S13. Get Scheduled, http://getscheduled.co.uk/

S14. GoAssign, http://www.goassign.com/

S15. QuickStaff, http://quickstaffpro.com/

S16. ShiftNote, http://www.shiftnote.com/

S17. STAFFomatic, http://staffomatic.com/

S18. EZShift Scheduling Software, http://www.ezshift.com/

S19. Shift Status, http://shiftstatus.com/

S20. 247Shift, http://www.247shift.com/

S21. Agendrix, http://www.agendrix.com/

S22. BSS Go, http://www.bss-go.com/

S23. FieldEZ, http://www.fieldez.com/

S24. Froster, http://froster.co/

S25. Humanity, http://www.humanity.com/

S26. LocalTable, http://www.localtable.co/

S27. MakeShift, http://www.makeshift.ca/

S28. QuickTask, http://quicktask.com/

S29. ScheduleBase, http://schedulebase.com/

S30. Schedulehead, http://www.schedulehead.com/

S31. ShiftPlanning, http://www.shiftplanning.com/

S32. Shiftworkz, http://shiftworkz.com/

S33. Sling, https://getsling.com/

S34. Workforce Management Software, http://www.qweek.com/

S35. ZoomShift, http://www.zoomshift.com/

S36. SubItUp, http://www.subitup.com/

S37. When I Work Scheduling, http://wheniwork.com/

S38. Planday, http://planday.com/

S39. Snap Schedule, http://snapschedule.com/

S40. Asgard Scheduling Software, http://www.asgardsystems.com/

S41. DRoster, http://www.kappix.com/

S42. Dienstag, http://www.dienstag-app.com/

S43. Deputy, http://www.deputy.com

S44. Findmyshift, http://www.findmyshift.com/

S45. TixTime, http://www.tixtime.com/

S46. PARIM, http://parim.co/

Appendix 4

ARTICLE 3: PARAMETERS TO CONSIDER WHEN IT-SUPPORING ROSTERING: A LITERATURE INTERDISCIPLINARY REVIEW

IT-supporting Self-rostering in a Complex Rostering Situation

Simon Sønderskov Iversen
Aalborg University
Department of Computer Science
sivers11@student.aau.dk

Daniel Ginnerup Kappers
Aalborg University
Department of Computer Science
dkappe11@student.aau.dk

Christian Sand Pilgaard
Aalborg University
Department of Computer Science
cpilga10@student.aau.dk

ABSTRACT
This article focuses on creating prototypes for an IT-system supporting Self-rostering, for a complex rostering situation. This is done through the use of a Human-Centred Design method, where two preliminary interviews were held, followed by a design session incorporating Effective Prototyping. Three prototypes implementing Self-rostering differently was the result of the design process, which ended with a focus group evaluation.

This process revealed that IT-supported rostering in general was welcomed positively, but most important that the prototypes were received with some scepticism. However, the participants of the focus group thought the concept of IT-supported Self-rostering as advantageous, if the proper ground-work was put into creating the system.

The evaluation revealed that IT-supported rostering in general was welcomed positively. The prototypes however were received with some some scepticism, because it was reasoned that it would be hard to create a system, able to support all rules present in the organisation. If possible to create such system, IT-supported Self-rostering was perceived as advantageous. Additionally, it was discovered throughout the study, that Self-rostering needs to be further defined in relation to IT-support

Author Keywords
Human-Centred Design; Rostering; IT-support; Effective Prototyping; Focus Group; Self-rostering; Complexity.

INTRODUCTION
Silvestro & Silvestro [1] describes three different Rostering Approaches, relevant in different rostering situations and with each their distinct advantages and disadvantages. Self-rostering, being one of these Rostering Approaches, is described as relevant in rostering situations of a maximum of 35 people, with what is described as a low Rostering Complexity.

Additionally, Mortley & Grierson-Hill [2] conducted a study on implementing analogue Self-rostering in a specific case, describing the advantages and disadvantages of Self-rostering, which are corresponding to the ones presented by Silvestro & Silvestro.

Silvestro & Silvestro [1] and Mortley & Grierson-Hill [2] only describe Self-rostering as an analogue rostering practice. Therefore, it was found interesting to investigate IT-support in relation to Self-rostering, as it was hypothesised that IT-support could reduce or eliminate some disadvantages associated with the Rostering Approach. Especially the limitation on Rostering Complexity and size was found of interest. This is done through a proof-of-concept study, as no literature was found, which covered the exact area of interest.

METHOD
As a method for the process of creating a proof-of-concept, it was decided to use Human-Centred Design, employing parts of the model within the ISO standard 9241-210:2010 for Human-Centred design for interactive systems [3], for which the design process was planned accordingly. Furthermore, Effective Prototyping [4] has been implemented within the ISO standard in order to supplement the following three activities of the model:

- Specify the user requirements
- Produce design solutions to meet user requirements
- Evaluate the designs against user requirements

Effective Prototyping prescribes iterations of three steps, corresponding to the three activities of the ISO standard.

The aim of this research is to investigate whether IT-support of Self-rostering would be viable for a complex rostering situation. This research should be seen as Basic Research [5] and takes the first steps in investigating different approaches to IT-supported Self-rostering in a complex rostering situation. Due to this scope, the intention is not to implement the prototypes, therefore the last activity of Human-Centred Design; Designed solution meets user requirements, is not conducted, likewise only three of six prescribed iterations of Effective Prototyping are implemented.

When conducting the activity *Produce design solutions to meet user requirements*, the prototypes were not based directly on the definition of Self-rostering as described by Silvestro & Silvestro [1], but rather on the most essential underlying mechanism of Self-rostering; a high level of employee influence. It was argued, that by increasing influence, many of the benefits of Self-rostering would follow.

Iteration 1

The first iteration includes five activities: Plan the human-centred design process, Understand and specify the context of use, Specify the user requirements, Produce design solutions to meet user requirements, and Evaluate the designs against requirements.

The first iteration had the focus of getting a thorough understanding of the problem area from a planner's perspective and gaining insight into the context of use to create User Requirements.

Plan the Human-centred Design Process

The process was planned as a Human-Centred Design process, incorporating Effective Prototyping in order to study whether IT-supported Self-rostering is viable in a complex rostering situation. The process is presented through the following sections.

Understand and Specify the Context of Use

Two interviews were held, each with their distinct purpose. In the first interview, all three authors were present along with the planner being interviewed. In the second interview two authors were present, as well as the planner. Both interviews were designed as prescribed in Kvale’s guidelines for interviews [6].

The first interview was with the planner of a residential accommodation for people with mental illnesses. The accommodation unit employs approximately 45 full-time employees along with part-time staff, a few interns and temporaries. Besides the amount of employees, the high level of rules ranging from laws to individual contractual agreement, adds to the complexity of the rostering situation. Even though the Demand Variability is Low and Demand Predictability High, an overall assessment of the Rostering Complexity in regards to the definition of Silvestro & Silvestro [1], is assessed as Medium. The interview consisted of a series of questions related to the rostering situation within the organisation, continuously relating the information to relevant theory. The purpose of this interview was to gain knowledge about the rostering situation specifically within the given organisation, which would also act as case for the development of prototypes along with an evaluation with employees of the institution.

The second interview was with the planner of a care centre for people with dementia, with approximately the same Rostering Complexity and size as the organisation in the first interview. The interview consisted of two phases. The first phase was for the interviewer to confirm the complexity of the rostering situation, and identify the rostering situation in relation to theory. The second phase was for the interviewer to get the planners opinion on Self-rostering on a conceptual level. The interview had two purposes. The first to gain knowledge about rostering in a similar rostering situation, in terms of Rostering Complexity and size. The second to gain insight into a planner's opinion about Self-rostering, for an institution with such complexity.

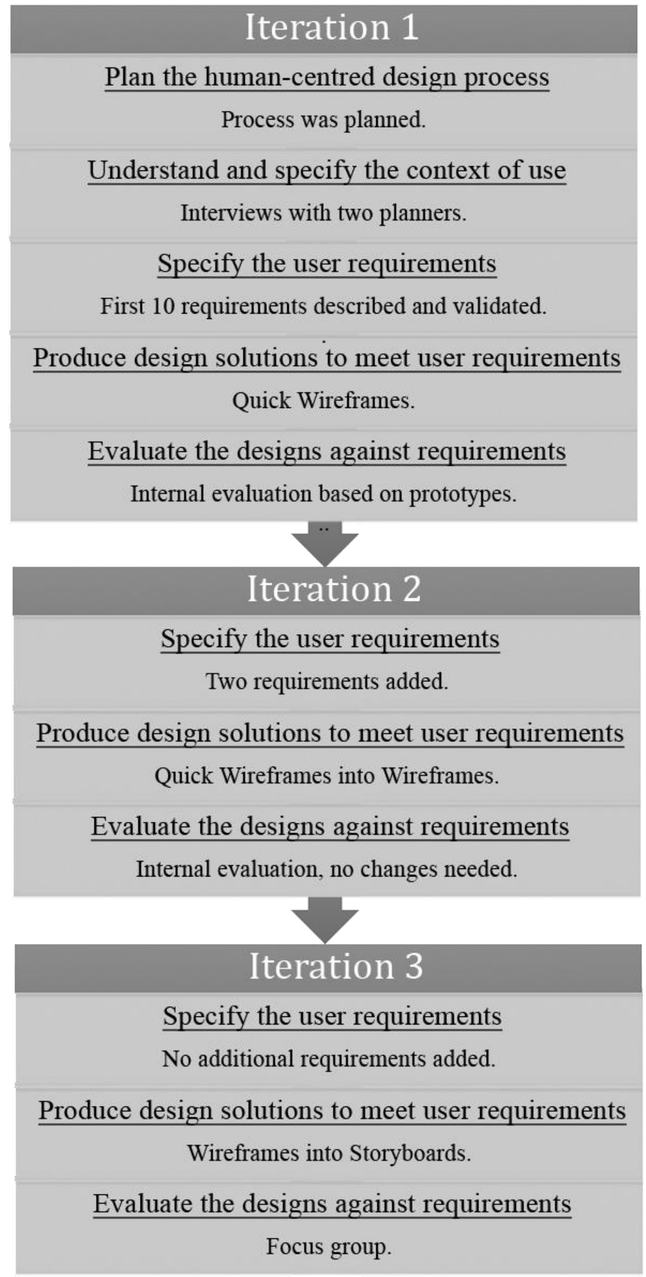

Figure 1. Process, based on Human-Centred Design and Effective Prototyping.

Specify the User Requirements

In the first iteration of this activity, the authors formulated the initial User Requirements jointly and filled these into the schema, as shown in Table 1.

The schema had been limited to only include three columns, instead of the total six Effective Prototyping prescribes [4]. The columns that were excluded are: Priority, Results, and Requested Change. Priority was excluded because it was predefined that all requirements had to be fulfilled, and Results and Requested Change were excluded because it was not allowed for the authors to access each other’s work. All User Requirements were initially discussed and written into the name column, and categorised according to the type of User Requirement. All requirements were then evaluated in relation to validity based on existing knowledge. The requirements could be validated through three criteria: Either through dictation from the case, through dictation from Self-rostering or through the scope of the study. The final schema as a result of all iterations can be found in Table 1.

Produce Design Solutions to Meet User Requirements
The authors conducted the first iteration of this activity by creating Quick Wireframes [4], in order to visualise a crude yet holistic expression of the requirements within the context of a functioning system. The program Balsamiq Mockups was used by all three authors to create Wireframes, in order to ensure that the Wireframes were not affected by the authors' drawing skills, and that the style of the prototypes would be similar. Additionally, it was not allowed for the authors to see each other's designs before the final evaluation, in order to avoid affecting each other.

Evaluate the Designs Against Requirements
The first evaluation activity was done internally among the authors themselves. The evaluation was done as a discussion, grounded in the sketches made from the previous activity.

It was determined that additional User Requirements were needed, one in relation to the handling of leave, and another related to the specific case.

Iteration 2
The second iteration includes three activities: Specify the user requirements, Produce design solutions to meet user requirements, and Evaluate the designs against requirements.

The second iteration had a focus on further evolving the designs, resulting in a better understanding of the holistic expression of the system.

Specify the User Requirements
The second iteration of this activity involved for the authors, to implement changes as a result of the previous evaluation, in collaboration. Two additional User Requirements were added, these are placed at the bottom of the schema, which can be found in Table 1.

Produce Design Solutions to Meet User Requirements
In this second iteration, producing the design solution involved for each author to evolve their respective Quick Wireframes into Wireframes. However, since User Requirements were added to the schema, each of these also had to be supported by the Wireframes.

Evaluate the Designs Against Requirements
The second evaluation was identical to the first. A discussion was conducted based on the existing wireframes. There was, however, found no need to change the existing User Requirements.

Name	Type	Validated
The prototype must support constraints to be applied on a general and individual level.	Functional	Yes (Case)
The prototype must support collective agreements, local plans and contractual agreements.	Business	Yes (Case)
The prototype must be web-based with focus on a desktop view.	Usability	Yes (Scope)
Users should be able to easily understand and use the prototype.	Usability	Yes (Case)
The prototype must support individual wishes and preferences.	Functional	Yes (Self-rostering)
The prototype should not require significantly more time than the current system.	Business	Yes (Case)
The users should have great influence on their own plan.	Business	Yes (Self-rostering)
The prototype should support the three departments.	Business	Yes (Case)
Users must be able to get an overview of their own work schedule for a minimum period of 4 weeks.	Functional	Yes (Case)
The prototype must be designed for multiple users.	Functional	Yes (Self-rostering + Case)
All shifts must be occupied 4 weeks in advance.	Business	Yes (Case)
The prototype must be able to support preferences for leave.	Business	Yes (Self-rostering + Case)

Table 1. Schema of final User Requirements for the prototypes, as prescribed by Effective Prototyping.

Iteration 3

The third iteration included three activities: Specify the user requirements, Produce design solutions to meet user requirements, and Evaluate the designs against requirements.

The third and final iteration had the focus of relating the User Requirements to an end-user perspective.

Specify the User Requirements

The evaluation from the previous iteration concluded that no changes were needed. The authors were aware of the nature of Storyboards being increasingly centred on the user's perspective, compared to the Wireframes. All User Requirements were discussed, in order to ensure the mutual understanding of each requirement in relation to a user's perspective. No changes to the schema was made.

Produce Design Solutions to Meet User Requirements

The third iteration of this activity is prescribed to focus on turning the Wireframes into Storyboards which will aid in focusing on the requirements from the end-user perspective. Three stories were chosen, which all Wireframes should depict:

- The user wants to see his/her own plan.
- The user wants to influence the future roster.
- The user wants to request for leave.

The three stories were selected based on the reasoning that the three would cover primary interaction for a Self-rostering system in the given context.

Evaluate the Designs Against Requirements

The final evaluation was conducted in Spring 2016, in the form of a focus group with four participants, all employees of the organisation used as a case, with a wide range of age and work experience. Two of the authors were present and acting as Moderator and Scribe respectively [7]. The focus group consisted of three steps, and was structured through Kvale's [6] and Greenbaum's [7] guidelines for focus groups.

The first step consisted of three phases. In the first phase the employees' current rostering situation was discussed. In the second phase employees were asked to individually write down five advantages and disadvantages of their current rostering practices. In the third phase the notes were used as an artefact [6] for further discussion.

The second step also consisted of three phases. In the first phase the moderator introduced the concept of self-rostering for the participants. In the second phase the participants had to write down their immediate thoughts on advantages and disadvantages of Self-rostering. In the third phase the writings were used as an artefact for further discussion on the concept of Self-rostering.

The third step consisted of two phases. In the first phase the storyboards were presented, and the participants were given handouts of each prototype, for them to study as artefacts for the discussion on IT-supported Self-rostering. In the second phase a discussion was conducted about each concept's functionality and viability as a potential system.

PROTOTYPES

Throughout the prototyping process, three prototypes were created; one by each author. All three can be seen in Figure 2-7. The three prototypes were all subject to the same User Requirements and were all based on different implementations of Self-rostering. The scope of the prototypes was to only include functionality relevant to the employee, not for an eventual administrator.

As the design sessions were finalised, it became evident that the authors' understanding of IT-supported Self-rostering was quite different. Therefore, the prototypes had substantial differences, which added an exploratory interest, in addition to the proof-of-concept.

In the following all three prototypes are presented, describing the unique characteristics for each system. The presentation will be followed by three subjects: System Role, Constraints, and Fairness, used to indicate the differences between the prototypes.

Prototype 1

The first prototype allows the user to have full control of which shifts to take. All shifts made available for a given period will be vacant, and remains so until an employee decides to occupy it. This can be seen in Figure 2 where there are remaining open shifts. The prototype is based on the expectation, that employees will have sufficient sense of responsibility, for the roster to be filled. Vacation is held through the act of not taking shifts. The system does not actively handle the User Requirement of shifts having to be occupied four weeks in advance.

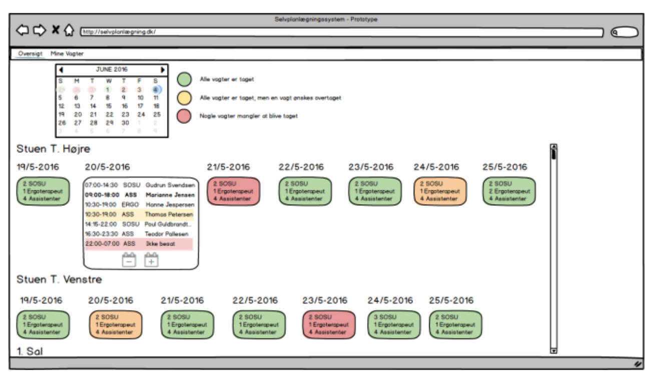

Figure 2. Prototype 1 - An overall view of the current roster.

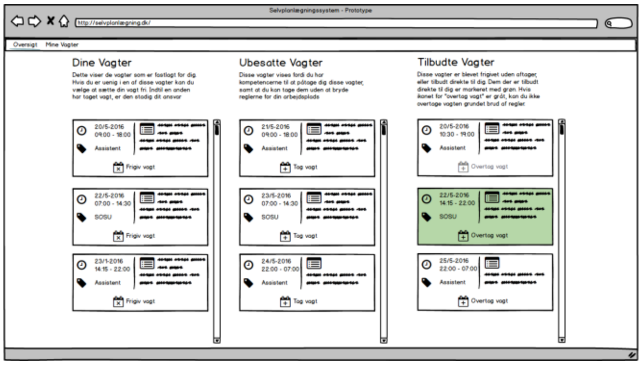

Figure 3. Prototype 1 - A view of how users can manage their own roster.

Prototype 2

The second prototype allows for the users to have full control of which shifts to work, if they are able to create a viable roster themselves. In case the roster is not viable at the deadline for publishing, the intention is for a planner to be able to put people on shifts. The planner must approve requests for vacation if they are requested in the published roster, which is defined as a separate request, as opposed to the previous system. In the system, each user can create a cyclic template for a chosen number of weeks, which will be repeated. The cyclic template can be seen in Figure 5, where the user is presented with a table where each column represents a given weekday, and the week number is presented for each row. Employees are able to change their shifts for individual dates, while not affecting the original template. This prototype is based on the idea of social responsibility and incorporates notices of broken constraints to promote social responsibility.

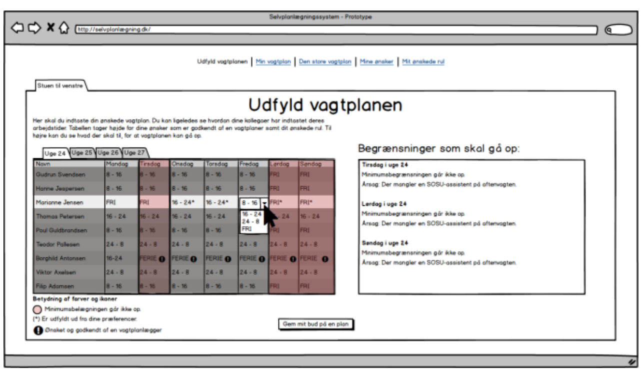

Figure 4. Prototype 2 - A view of how users can manage their shifts for the next roster.

Figure 5. Prototype 2 - A view of how users can make changes to their template.

Prototype 3

The third prototype does not allow for employees to have control of their shifts directly, instead it is based on objective generation of a roster based on each users' preferences for work and leave. It is seen in Figure 6, how a user enters preferences for specific shifts. As the roster have to be published in advance, the system sets a deadline for users to change preferences, after that the roster will be created and is considered final. If employees seek changes to their roster after the system has created a roster, they will have to ask for another employee to cover their shift, or request for leave. Requests for leave must be approved by a planner.

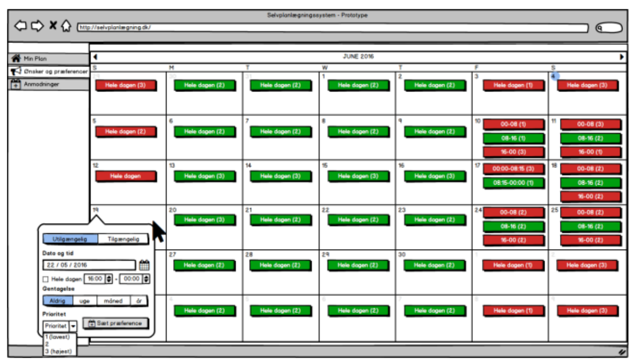

Figure 6. Prototype 3 - A view of how users can manage their preferences for rosters.

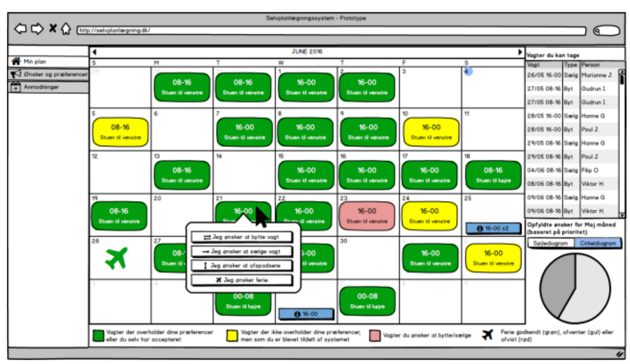

Figure 7. Prototype 3 - A view of the published roster.

System Role

Among the three systems, the appointed role of the system itself is very different from each other, scaling from facilitation of occupying shifts, to an intelligent rostering system. The systems roles are described using metaphors [8], as shown in Table 2.

Prototype	Role
Prototype 1	Facilitator
Prototype 2	Supervisor
Prototype 3	Coordinator

Table 2. The prototypes and their roles.

The first prototype is a rather simple tool with the role of a Facilitator, meaning that it facilitates an overview of which shifts are vacant and which are occupied, as seen in Figure 2. The prototype supports a high level of employee empowerment, as the employees can freely choose between shifts within their job title and applicable rules. This comes at a price of stability, as it does not provide any means of forcing the employees to take shifts, therefore issues can arise if not enough employees decide to takes shifts, which will result in the roster not being viable.

The second prototype's role is more so of a Supervisor, making sure the employees manage their shifts. If the employees are not able to create a viable roster themselves within a certain deadline, the system will call upon a designated planner to solve broken constraints.

The third system takes the role of Coordinator. The system will collect all preferences before creating the roster, and then simulate each employee objectively to argue among each other, and use the results of that to create a viable roster. An example roster is presented in Figure 7.

Constraints

The concept of constraints, is defined as a set of restrictions within the system which limits the rostering problem. Constraints has been implemented into every prototype, but not always explicitly through the design. This section will describe the prototypes in relation to constraint feedback and placement of responsibility for upholding the constraints, as illustrated in Table 3.

Prototype	Feedback	Responsible
Prototype 1	Medium	System + Employees
Prototype 2	High	Planner + Employees
Prototype 3	Medium	System

Table 3. Prototypes related to constraints, in terms of feedback and responsibility.

The first prototype technically prohibits the user from taking actions that will break constraints related to acquiring a shift, therefore the System is marked as responsible. The users themselves however, are responsible for the remainder of the constraints, and are therefore also responsible. All constraints in the system are communicated visually, in terms of greying out functionality that is not allowed or colouring areas related to a broken constraint; the constraint is not explicitly described however, which places the feedback level at Medium. Greyed out functionality is seen in Figure 3, where the option to take a shift is greyed out.

The second prototype does not technically prohibit the user in any way. The prototype instead uses colouring and writing to visually inform all employees of which constraints are broken, giving a High level of feedback, but placing the responsibility to solve broken constraints upon the Employees. Examples feedback can be seen in Figure 4. If the employees do not manage to fix the constraints themselves, the Planner will be informed by the system, to take responsibility of fixing the broken constraints.

The third prototype enforces weighted constraints when creating the roster, meaning that it will take preferences into consideration, if they can be fulfilled without breaking constraints. Therefore, the System is fully responsible for upholding the constraints. After the roster is created, the system will inform each employee of broken constraints for their own roster, through colouring and statistics, placing the feedback level for the system at Medium. An example of the feedback given, can be seen at Figure 7.

Fairness

The fairness that theoretically could be implemented through organisational procedures outside the boundaries of the system, is not taken into consideration. The following will describe each prototype in relation to the objectivity of the fairness that can be implemented, along with the placement of responsibility for implementing fairness. The categorisation of the systems can be seen in Table 4.

Prototype	Objectivity	Responsible
Prototype 1	Low	Employees
Prototype 2	Medium	Planner + Employees
Prototype 3	High	System

Table 4. Prototypes related to fairness, in terms of objectivity and responsibility.

In the first prototype, the employees have the power to take, refrain from taking, and leave shifts, thereby increasing or decreasing fairness for the entire roster. This places the entire responsibility of fairness upon the employees. The objectivity of fairness will therefore also be low, as it is expected that some employees will attempt to improve their own situation over others'.

The second prototype also allows for the users to take, refrain from taking, and leave shifts, as long as the roster does not break constraints. Therefore, most of the responsibility for fairness is placed upon the employees. If the roster is not viable, the planner will take over, and the responsibility of fairness changes. As the planner will be able to change shifts for all employees, the subjective fairness can come closer to an objective assessment.

The third prototype only takes preferences into account, but makes all decisions related to the final roster. Therefore, the responsibility for fairness will be placed upon the system, and the level of objectivity will be high, as the system will weight each employee equally.

RESULTS

The final evaluation of the prototypes, in the form of a focus group, resulted in data concerning two areas: Self-rostering as a concept, and IT-supported Self-rostering in relation to the prototypes.

Self-rostering

The participants of the focus group were presented to the concept of Self-rostering and seemed rather perceptive of the consequences of the concept. The participants often related Self-rostering to their own rostering situation.

The highlighted advantages were especially: The added influence on the roster, the fact that the roster was not dependent on a single planner, and the idea of a shared social responsibility for the roster. The participants also reflected that a shared responsibility would mean that no single person could be blamed for the final roster. Another advantage highlighted was the ability to adjust rosters in accordance with new employees arriving to the team. The participants compared it to their current situation where new arrivals have to take over a leaving employees roster, as they have a cyclic roster, which is reworked rarely. The flexibility of Self-rostering was also reflected in relation to existing employees, and it was concluded that it would be easier to change one's work schedule in accordance to changing needs in employees' private lives.

One concern for Self-rostering was related to the idea of a shared responsibility and added influence, as the participants did not expect all employees to be able to uphold such responsibility. This concern was enhanced with an added concern of inequality in the taken shifts, as it was argued that some employees would be more flexible and prone to taking more shifts than others, where others will make sure to only serve their own interests instead.

In relation to this inequality and problems with shared responsibility, another concern is that all rules must be upheld, and they were very doubtful of their ability to uphold these, if the roster was a shared responsibility.

The final concern was in relation to a lack of predictability of the roster, as the participants were used to having their cyclic roster, meaning they could always make an estimate of their work schedule several months in advance.

Prototypes

The general attitude towards the prototypes presented in the focus group, was of interest, but also scepticism. It was evident that the participants had a hard time understanding the idea that the system could take all organisational rules into consideration, without a planner having to interfere. This resulted in a scepticism for the implementation, as the participants reasoned it would require a vast and thorough amount of preliminary work in order to replace the planner's function and knowledge.

Throughout the discussion of the prototypes, three subjects emerged: System role, Constraints, and Fairness

System Role

A discussion of the right role for the system, was undertaken during the focus group. The discussion was related to how much flexibility and responsibility a given employee was able to uphold. The discussion revealed a tendency towards the participants finding a high amount of responsibility unrealistic to uphold, for some employees. This meant that the participants found the Supervisor and Coordinator role more satisfying than the Facilitator role, as they presented countermeasures for the roster not being viable.

The participants generally expressed a liking towards the freedom of choosing between one's own shifts, however their assessment was that realistically, a planner or the system had to be able to take decisions when the employees were not able to create a viable roster.

Constraints

All three prototypes handle the concept of constraints for the roster differently. The participants expressed a lack of trust towards the first prototype due to the it only supporting a Medium amount of constraint feedback to the user. It was reasoned by the authors, that this was in conjunction with the placement of responsibility upon Employees. The second prototype was perceived more positively, as the feedback level is High, but also because the placement of responsibility for constraints are shared between Planner and Employees, helping to ensure they are upheld. The third prototype has the same level of feedback as the first, but because the placement of responsibility is only on the System, they find the Medium amount of feedback less concerning.

Fairness

For the first prototype, the participants had similar concerns to Self-rostering in general. They found the placement of responsibility for fairness on Employees disadvantageous, because the expectation was that employees would not be able to disregard their own priorities, in favour of fairness. Related to the second prototype, the concerns were diminished, as it was explained that a Planner could make changes in order to make the roster viable, if the employees were not able to do it on their own, meaning the Planner and the Employees would both have the responsibility for fairness. The third prototype was also met with slight critique, however the placement of responsibility upon the System, joined with the explanation of the High objectivity of an algorithm, increased the participants' liking for the system. The concern that remained was the fact that the system does not take personal issues, into account the same way a planner does.

RELATED WORK

A single similar study has been identified, reporting results on the implementation of a Self-rostering IT-system.

Ball [9] describes a study where an IT-system for Self-rostering has been implemented. A total of approximately 77 employees were distributed across three hospital wards. No further description of the rostering situation in terms of size or Rostering Complexity is given, beside that the system cannot account for grade mix. The rostering process for the IT-system consists of four steps. In the first step, employees are required to enter their preferences for work, while they have an option to veto for days they do not want to work. In the second step, the IT-system creates a roster, aiming to combine the employee preferences, while outputting the coverage of shifts. In the third step, the employees are asked to volunteer to cover shifts that are not already covered. In the fourth and last step, the IT-system forces employees upon shifts. The system has a time banking feature, where an employee can carry over a credit or debit of hours from one roster to another, accumulated by working more or less hours than contractually agreed.

The reported results are that the longer employees have used their old system, the more hesitant they are towards the new one. The time banking feature was welcomed very differently; some enjoyed the flexibility that followed the feature, while others did not like the idea of working more hours without getting paid, and owing hours to the organisation if working less. In general, the results revealed positive feedback on the system, especially based on the level of freedom the system grants, and the fact that this improves the work-family balance. The article also reports a few individuals describing an increased motivation performance in relation to their work.

DISCUSSION

A definitive conclusion to the proof-of-concept study within this article is not possible, as an actual implementation has not been conducted. The study however, shows strong tendencies towards IT-supported Self-rostering being viable in complex rostering situations, it would not normally be viable within. If Future Work was conducted it should be towards creating a functional prototype, followed by an implementation into a context with a similar size and Rostering Complexity to the one described in this study.

A proof-of-concept study was conducted through this article, however as shown in related work, there has already been implemented an IT-supported Self-rostering system in several hospital wards [9]. Our article, and the one by Ball [9] are relatable, as both aim to prove the viability of Self-rostering, through the use of IT. However, the article by Ball does not focus on the size and Rostering Complexity, and no description of either is present in the article. The design and implementation process for the study is also hard to determine, as it is only sparsely described. It was therefore decided that a proof-of-concept study, related to a specifically elaborated case, adds to the relevance of the study, as it would be easier to replicate.

Silvestro & Silvestro [1] describe three Rostering Approaches, in relation to an organisational structure. The descriptions, however, does not relate these Rostering Approaches to IT-support, and are therefore only applicable to an analogue rostering processes. In order to apply Silvestro & Silvestro's definition to an IT-supported rostering situation, Self-rostering has been interpreted into a broader perspective. This is done by incorporating the most essential underlying mechanism of Self-rostering when creating the prototypes; increased employee influence. While the prototypes were designed to have high levels of employee influence and added flexibility, the issue of the systems partaking, either actively or passively, in the rostering process have been largely ignored. Comparing the prototypes to the classic definition of Self-rostering, it becomes apparent that the prototypes, to varying degrees, may not comply with the definition. One of the main challenges is to define which kind of role the system is allowed to have, as any kind of decision making or guidance can affect the fundamental dynamics of the Rostering Approach.

When conducting a study where users are included, there is a risk of bias in the form of research pleasing. This is also a risk within this study, as the participants of the focus group might have attempted to answer in accordance with our expectations or hopes. It has been attempted to countermeasure this, through the use of methods including countermeasures of bias in interviews and focus groups [6-7]. It was observed that the participants were comfortable expressing both positive and negative sides of the presented prototypes.

REFERENCES

1. Rhian Silvestro and Claudio Silvestro. (2000). An evaluation of nurse rostering practices in the national health service. *Journal of Advanced Nursing* 32, 3:525-535.
2. Val Wortley and Lesley Grierson-Hill. (2003). Developing a successful self-rostering shift system. *Nursing Standard* 17, 42:40-43.
3. ISO/TC 159. (2010). Ergonomics of Human-System Interaction -- part 210: Human-Centred Design for Interactive Systems.
4. Jonathan Arnowitz, Michael Arent and Nevin Berger. 2007. Plan your prototype. *Effective prototyping for software makers (interactive technologies)* (1st ed.), Dawnmarie Simpson and Asma Palmeiro (Eds.). Elsevier Inc., San Francisco, CA
5. J. L. Wynekoop and S. A. Conger. 1990. *A review of computer aided software engineering research methods*.

Copenhagen, Denmark.: Proceedings of the IFIP TC8 WG 8.2 Working Conference on The Information Systems Research Arena of The 90's.

6. Steinar Kvale and Svend Brinkmann 2009. Interviews: Learning the craft of qualitative research interviewing (3rd ed.) Sage.

7. Thomas L. Greenbaum. 1998. *The handbook for focus group research* (2nd ed.) Sage Publications Inc.

8. Lars Mathiassen, Andreas Munk-Madsen, Jan Stage and Axel P. Nielsen. 2000. *Object Oriented Analysis and Design* (pp. 29-40). Marko

9. Jane E. Ball. 1997. Shifting the control: Evaluation of a Self-Scheduling Flexitime Rostering System. *European Nurse Research,* 2, 2: 100-110.