HAAGA-HELIA
ammattikorkeakoulu

Context-aware design: Developing an adaptive and automated newsfeed plug-in.

Michaël Dayen

Bachelor's Thesis
Business Information Technology
28.08.2015

Table of Contents

Table of illustrations

Acknowledgements

I would like to thank my school's advisors for this thesis:

Dr. Amir Dirin, my thesis coordinator in Haaga-Helia, Finland, for the opportunities he gave me throughout the year, allowing me to explore a fascinating new concepts and for his advices and support. Without his trust, this research wouldn't have been conduct and my other recent projects would not have been started.

Dr. Florian Evequoz, my thesis coordinator in HES-SO Valais, Switzerland, for his help and support and availability during the thesis process.

I would show my gratitude to Mr. David Wannier, head of BIT in HES-SO Valais, and Mr Jarmo Peltoniemi, Head of BIT in Haaga-Helia, for their effort to make this exchange year possible and allow me and my colleagues to be the first persons to experience it.

I show my appreciation to Mr. Rapahël Sculati, who worked with me on mHealth project and to many other project. We will certainly continue to produce many more quality projects together in the upcoming years. And finally, to all the other students involved in the mHealth project.

Abstract

Authors Mr. Michaël Dayen	**Group** mHealth project
Title Context-aware design: Developing an adaptive and automated newsfeed plug-in.	**Number of pages** 44

Supervisors
Dr. Amir Dirin - School's advisor of the thesis in Finland
Dr. Florian Evequoz - School's advisor of the thesis in Switzerland

Context-aware mobile application design and development are becoming popular nowadays. The Context-awareness solution is applied in mobile applications mainly to anticipate users' needs and deliver appropriate content based on users' contexts.

This paper reveals different approaches to design automation to build our framework. We define the blocks of our context-aware design framework (CADF).

Furthermore, we demonstrate and assess our architectural solution in a mobile health application prototype to demonstrate a proof concept.

The main aim of the CADF prototype within a healthcare mobile application project (mHealth) is to reduce the need for users to actively input information in the system by proposing context-aware content to the users based on their work-habits and the context of use (e.g. the environment where the application is in use).

Context-aware design makes use of several different ICT technologies, such as Design-Pattern, Object-Oriented HTML & CSS, Process Automation, UX best practices, Web Frameworks and Data-driven design to dynamically generate a user interface for a random amount of information collected from the back-end. This paper is a first-step to what might become a more efficient approach to web and user-experience design automation.

Key words
Usability, user experience, user-centred design, context-awareness, object-oriented design, process automation, con-text-aware design framework, CADF.

1 Introduction

With the rise of Artificial Intelligence (AI) technics, outstanding web-development frameworks such as Bootstrap or LESS to give programmers even more flexibility during the development of new applications, it is likely that we will soon make use of websites and mobile application that design themselves without any human support. (Gevarter, 1987; Hutchison, 2005; Twitter, 2014)

In 2015, a few companies are launching their first version of AI web designers and there are very high expectations for their launch. (Foster, Kesselman, & Kesselman, 2004)

In this paper, we review the related literature on context-aware design and reveal different approaches to problems such as: (1) programming a machine with all the good UX practices that a designer learns during their career, (2) by analysing stylesheet data to apply machine learning and automatically generate new designs.

We applied the context-aware design framework (CADF) to automatically design user interfaces for a mobile information system used by hospital staff called mHealth as a proof of concept. (M. Dirin, Dirin, & Laine, 2015)

In close collaboration with Mr. Amir Dirin and Dr. Florian Evequoz, this work is also presented in an academic article format, for research publication purposes.

2 Objectives and Scope

The objectives of this research paper are:

1) Producing a context-aware framework to automate the design workflow of a screen, to present our design and development automation of mobile user interface design based on user context, dynamically collected data, and predicted services. Therefore, to generate the target user interface the CADF framework generates the contents such as the images, text and numerical values. Moreover, the user interface for the target application is generated based on the user's profile and the context in which the user interacts with the application, hence the application content and the user interface for two different users may vary according to their environment.

2) Through a review of the literature we study the most feasible alternatives to design appropriate approaches to automate the web-design in the back-end with less usability errors in the generated user interface design and provide the most probable required content to the target users.

Finally, we assess our findings by applying them to the mHealth project. We evaluate how our automated design approaches results in a context-aware design framework for different nurses at the hospital. The scope of this paper is mainly concentrated on the intersection between web-design and the context-awareness.

For academic verification, a mobile screen will be displayed correctly depending on the content and the context. The documentation will answer to the objectives questions, will picture the current situation in user-experience and explain what can be improve by generic display and design mining. The schemes will include the project architecture, the user-experience architecture, mock-ups, data-mining scheme and further more details, depending on the findings during the project, as it has not been explored yet in academic research.

The writer of the thesis will learn about user-experience in depth, middleware and web-service technologies and will pick the most efficient data-transportation method to connect the back-end of the architecture to the front-end, will use machine-learning algorithms for design mining and may have to learn few scalability tools for web-design and web frameworks.

3 Methods & Background Data

3.1 Methodology

In this paper, we try to observe a large scope of concepts before establishing a workflow. As this paper is our first essay in design-mining and context-aware design, we will consider multiple approaches and try to learn where engineers and researchers are focusing their effort to produce automated content and flexible code when it comes to web-design. We review the state of the art of three distinct approaches to design automation (UI/UX best practices, Design-mining & Object-oriented/dynamic design), we combine the approaches and construct a model based on the findings of each field, we implement this model within mHealth project and we assess and discuss the results.

3.2 Background data

This paper is based on review of the literature on design automation, human-machine interaction (HMI) and creativity automation. The review collected data mainly from papers published in design automation & context-awareness from 1994 to 2014 and technical documentation from web frameworks.

This paper also refers to a journal specialized in context-awareness that experiments with ways to create efficient user-centred and context-aware mobile applications. This paper will be part of the journal paper on context-awareness. Therefore, we implement our proof of concepts within the mHealth project.

The mHealth project is one of our core background data for this work. As it comes at the end of the context-aware workflow that mHealth aims to test, we based our work on prototypes that we develop during early stage of mHealth prototyping.

3.3 The mHealth project

The mHealth project was started to find a solution to provide an ergonomic and user-friendly mobile application design for hospital staff in their use of technology tools. The project objective is to reduce the need for users to actively input information in the system. The HMI is needed to display work information by proposing personalized content to the user, based on his behaviours and his localization within the hospital. The application was designed and developed in close collaboration with healthcare professionals. The mHealth project replaces existing manual work with an alternative: a context-aware mobile application. The multi-disciplinary design and development team composed of teachers and students from Haaga-Helia University of Applied Science, Ajou University in South Korea, HES-SO University of applied science of Western Switzerland and UXit Consulting.

The data used in this research are provided within a healthcare application. Medical records and patient data are used to build our proof of concept.

The requirements of the project had been discussed between the sponsor of the project and the project managers. A first prototype was developed during fall 2014 and was assessed by the end-users. At the time of writing, the prototype is available in Finnish and English. Four Haaga-Helia research students are writing their thesis on mHealth, and two other Haaga-Helia students are working on the design of the prototype.

Once implemented, the advantages of this project will be numerous. From a business-process perspective, the hospital will increase staff efficiency by simplifying the user-experience of their information system. Therefore, they will have more time to focus on their primary tasks because the time used to gather information will be significantly reduced. Furthermore, from an academic research perspective, the project explores and recommends alternative applications for context-awareness in information systems. We have identified three main fields of application: (1) in-building geo-localization, (2) understanding of user behaviour for better machine learning, and (3) design-mining to create generic user-experience.

3.3.1.1 Conceptual Architecture

The following figure shows us the conceptual architecture of the mHealth project. The components included in the red square are part of this research paper, focused on design production.

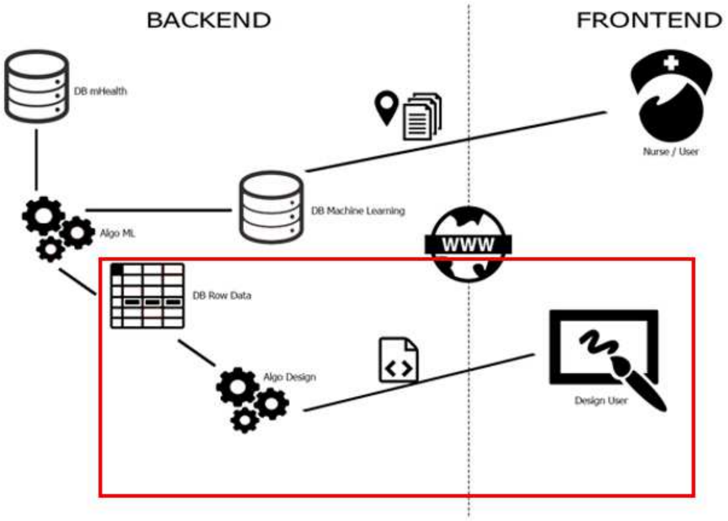


Figure 1 Conceptual architecture

3.3.1.2 Context-awareness in mHealth

Our main goal in mHealth is to design for the mobile context. Our application is built to satisfy the primary need of hospital staff: getting patient information whenever it is relevant. This is the global context of mHealth we also work on other context behaviour that affect the usability of the mobile information system. The location or physical context of use is handle by NFC tags that allow the device to know where the user is located. The device of access is a huge factor in the design process of our research. In this paper, we set up programming rules to adapt content to the device, maximize the screen use and display information in a user-friendly approach.(Fling, 2009)

For the development of the mHealth application, we have applied User-Centred Design (UCD) Principle and mLUX mobile application framework . The mLux framework consists of three layers: 1. the role-players 2. the process 3. the context of use. Figure 2 presents the overview of the mHealth application development framework. (Abras, Maloney-krichmar, & Preece, 2004; A. Dirin & Nieminen, 2015)

We continued to apply the UCD method to enhance the automated web design to mHealth application.

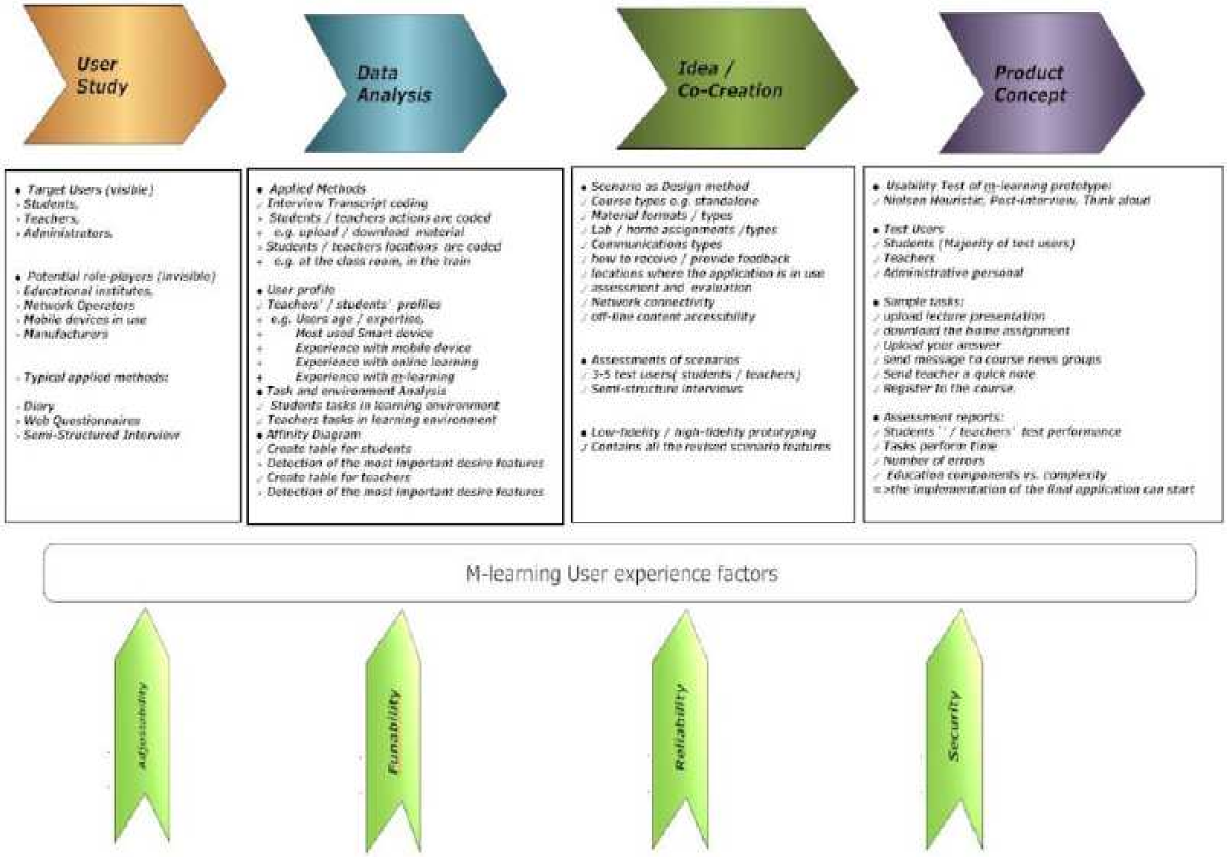

User Study
Data Analysis
Idea / Co-Creation
Product Concept
• Target Users (visible)
- Students,
- Teachers,
- Administrators.
• Potential role-players (invisible)
- Educational institutes,
- Network Operators
- Mobile devices in use
- Manufacturers
- Typical applied methods:
- Diary
- Web Questionnaires
- Semi-Structured Interview
• Applied Methods
✓ Interview Transcript coding
- Students / teachers actions are coded
+ e.g. upload / download material
- Students / teachers locations are coded
+ e.g. at the class room, in the train
• User profile
✓ Teachers' / students' profiles
+ e.g. Users age / expertise,
+ Most used Smart device
+ Experience with mobile device
+ Experience with online learning
+ Experience with m-learning
• Task and environment Analysis
✓ Students tasks in learning environment
✓ Teachers tasks in learning environment
• Affinity Diagram
✓ Create table for students
- Detection of the most important desire features
✓ Create table for teachers
- Detection of the most important desire features
• Scenario as Design method
✓ Course types e.g. standalone
✓ Material formats / types
✓ Lab / home assignments /types
✓ Communications types
✓ how to receive / provide feedback
✓ locations where the application is in use
✓ assessment and evaluation
✓ Network connectivity
✓ off-line content accessibility
• Assessments of scenarios
✓ 3-5 test users(students / teachers)
✓ Semi-structure interviews
• Low-fidelity / high-fidelity prototyping
✓ Contains all the revised scenario features
• Usability Test of m-learning prototype:
✓ Nielsen Heuristic, Post-interview, Think aloud
• Test Users
✓ Students (Majority of test users)
✓ Teachers
✓ Administrative personal
• Sample tasks:
✓ upload lecture presentation
✓ download the home assignment
✓ Upload your answer
✓ Send message to course news groups
✓ Send teacher a quick note
✓ Register to the course.
• Assessment reports:
✓ Students ' / teachers' test performance
✓ Tasks perform time
✓ Number of errors
✓ Education components vs. complexity
=>the implementation of the final application can start
M-learning User experience factors
Adjustability
Funability
Reliability
Security

Figure 2 MLUX development framework

3.3.2 **Related researches**

This paper is part of a global study on context-awareness in mobile environment. The implementation of the mHealth project was based on mLux: Moreover, other mHealth applications have merged several advanced technologies such as user profile and behaviour analysis using data predictions. Furthermore, we have applied technologies such as NFC and WIFI triangle for location coordination analysis. Therefore, by embedding the mentioned technologies with the proposed web-design approach we managed to generate a fully context-aware application that the user interface is constructed over the air. (Benyó, Vilmos, Kovacs, & Kutor, 2007; Dirin & Nieminen, 2015; Garamszegi, 2011; Yang et al., 2008)

All the design-mining related part of the paper are based on an award-winning HMI paper Webzeitgeist that gives the guidelines of design extraction.

Web based application development is one of the most evolving form of Information Technology. Therefore new development models and frameworks along with new programming languages are recommended to facilitate the work of full-stack, front-end and back-end for developers. Hence, web application development has become easier and simpler than before as the development tool become user friendlier, which required less advanced programming skills. Furthermore, the costs of the application production are still relatively high for small businesses. However, academicians and practitioners are continually developing new tools. For example, start-ups such as the Grid have tried to find solutions to automate the design process of websites. In this paper, we explore different approach to the problem to find which tools can be used to build autonomous designs. (Laine, Shestakov, Litvinova, & Vuorimaa, 2011)

3.3.2.1 Design automation approaches

As the production of web-design and mobile design are largely created by teams of web-designers, front-end or full-stack developers, there is a wide range of approaches to automate the design phase of an application. As this study is a starting point and aims to explore a large scope of concept, it will be difficult to fully automate the whole design process. It is difficult to be as good as a human brain in the creative process, but the machine can help the development team create faster with a data-driven strategy.

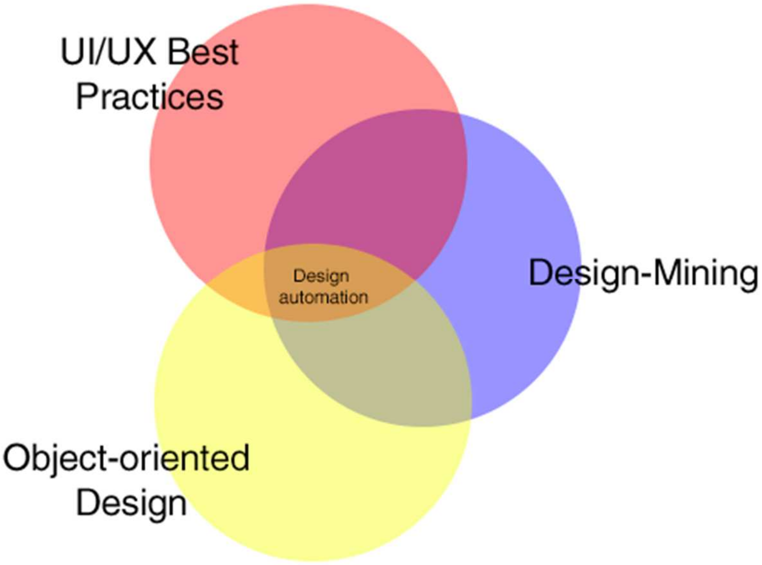


Figure 3 Automated Web-design approaches used for our model

In this section, we explore three approaches to automate web-design; we establish the strength and weaknesses of each approach and define a model based on the findings:

1. UI/UX Best Practices

2. Object-oriented design

3. Design-mining & Machine-learning

3.3.2.1.1 UI/UX Best Practices

In our first research approach, we tried to understand how front-end developers and web-designer use best practices and inspiration to build their own user-interface and user-experience. We also review standardized web-frameworks to integrate them into our development and in the workflow of the automation. The strength of this approach is to integrate to a system all the knowledge a real web-designer would have to learn to be efficient.

In a human-machine interaction perspective, we consider the computer in this model a colleague and part of the development team, based on Todd Lubart finding in his research on computers and human synergies in the creative process. Lubart considered that a computer can be creative and a colleague, can study the overall trends in web-design and propose a real help (design) to the designers. (Lubart, 2005)

As highlighted by S.R Herring the inspiration is driven by examples, as the computer can scan and find some similarity with past projects, it could reduce the time used in research before beginning to design screens, in web-development. There are many resources that are used by front-end developers and designers. Pttrn.com, is a great resource of examples for mobile development. Awwwards.com publishes the latest and best websites, in a design perspective, and rate them. The machine should be able to be inspired by those kind of websites and create great insights on this analysis to help the development team to identify current trends and avoid using components that tend to disappear. (Awwwards.com, n.d.; Herring, Chang, Krantzler, & Bailey, 2009; Pttrns.com, n.d.)

Examples are a big part of the method use in our mHealth implementation, rather than automate everything; we were trying to follow some standards in content presentation. By creating layouts and reusing them in different contexts, we were able to create screens that were familiar to most users.

This paper adopts List and Table-based layout design patterns, explained in Designing the Mobile User-Experience. We consider both of the approaches because of the responsiveness to the device. As explained in the book, Listlayout is suitable for any small devices (e.g. smartphones) and table layouts are more usable for larger device (e.g. tablets or laptops). Design Patterns are not standardized or defined in academic research but are the building blocks of our UI/UX best practices approach. (Ballard, 2007)

Bootstrap is an open-source framework and market standard that handles screen size in web development and provide a minimalistic design for a wide range of HTML5 elements. Standardized by the W3C consortium, Bootstrap is used in most modern websites. Bootstrap arranges the content based on the screen-size and therefore is a very strong framework for multi-platform development. The framework organizes layout in a maximum of 12 columns. For smartphone, tablet and laptop screensize, front-end developers can declare how many of those lines should be used by a specific class. It allows them to organize content into a grid. (Twitter, 2014)

However, one of the problems with best practices is the lack of originality in the display. By following very standard rules, designs will appear very similar to each other. A good example of use that is restricted are template-generated websites. They tend to be impersonal. In addition, there are tasks that humans can do better than machines and vice-versa. It is difficult to program a machine to be creative. On the other hand, computers are able to execute and calculate a lot faster than humans.

3.3.2.1.2 Object-oriented and dynamic Design

Object-oriented programming (OOP) handles a wide range of computer science problems. By allowing code to be scalable and modular, OOP is a very powerful tool for automation. In front-end programming, OOP is used in a wide range of frameworks. In this study, we focused on frameworks that deals directly with style sheets: (Oriented, Programming, & Oo, 2001)

For example, the @media queries are used to deal with screen widths dynamically by specifying an exact number of pixels. Media queries are built to deal with minimum-width and maximum width values and screen-types. Widely used since the rise of smartphones and tablets, media queries are very modular and allow developers to approach the problematic of responsiveness, dynamically. (Marcotte, 2010)

LESS is a pre-processing language used to add variables to CSS. One of the feature of the language is the LESS rendering into CSS code before launching it or on server-side when the browser is processing. Based on Javascript, the LESS files are compiled by Javascript files, to be rendered in a CSS file. The encapsulation of CSS code is another asset of LESS.

It allows front-end developers to declare a hierarchy of objects. Furthermore, SASS is another pre-processing language and a LESS alternative. SASS is mainly used with Ruby libraries, which makes it less convenient to implement in the mHealth application. LESS was our choice for mHealth because it can be ported to android or iOS code via multi-platform tools like PhoneGap. The entire front-end of the mHealth application is written in JQuery , Javascript and LESS because PhoneGap supports only JavaScript based web-applications. (Adobe, 2013; Eppstein, n.d.; W3Schools, 2014; Wikipedia, n.d.)

3.3.2.1.3 Design Mining & Machine Learning

Statistics are very important and largely apply in software automation and artificial intelligence products. By using metrics, it is possible to automatically produce very precise results. Our idea with studying Design-Mining and Artificial intelligence to create websites is based on recent datasets that have been produced by academic Labs in Stanford, M.I.T and Google. The startup scene is also very interested in AI designers. Furthermore, few companies are well funded enough to produce new ways to create websites and the hype around those projects is significant. Many customers are waiting for a new solution that will be more flexible than template-based website generators and expensive web-development teams. For example,

Webzeitgeist is a Stanford and M.I.T. award-winning research paper which is based on design mining. Webzeitgeist elaborated how to analyze web pages and create a dataset of DOM elements. Moreover, Webzeitgeist analyzed more than twelve thousands webpages to produce a gigantic dataset of DOM elements. While we were considering the option of using it in our solution, the datasets was not accessible for the public anymore. In their research publication, Webzeitgeist published few data visualization of their findings. For example, a 2D heat map of a webpage was produced to show where a web-element was most likely to be displayed on a webpage. (Kumar, Satyanarayan, & Torres, 2013; Talton, Ahmad, & Klemmer, n.d.; W3Schools, 2014)

In its Web Authoring Statistics, Google classifies and digs into the use of specific HTML5 components. The statistics outline which component are the most used and therefore enable designers to choose wisely their components, based on the market trends. (Google, n.d.)

The next step, in academic research about Artificial Intelligence designers, is to use those dataset to their full potential to find out what are the most common ways to design a web element, a web page or an entire website. That way, the consumer will be very familiar with the layout of an AI-generated design. Another advantage of metrics and machine learning is the evolution of design with time. It is interesting to consider a design that can evolve by following the current market trends.

An important weakness of data-driven design is that the designs are produced by machine-learning algorithms and therefore may lack originality, since they are learned from a large collection of pre-existing designs.

We may consider building machine learning on top of those datasets to try to propose designs, in further research. Extraction methods, used in Webzeitgeist research, are very interesting in our perspective of getting information automatically on a given source.

3.3.2.1.4 Merging the Approaches to create an automated model

After revising the literature around different design automation approaches, we decided to merge the three approaches to create a model that will be more adaptive than the current front-end programming technics.

As, in the given time, it would be impossible to construct an entire machine-learning engine to extract data from all around the web and design an entire website by artificial intelligence, we created a model to construct context-aware designs. In our use case, the screen displays a nurse newsfeed from a healthcare system. It shows information that is meaningful to the user in a given context. The study of user behaviour and information retrieval were part of another related study. In this paper, we concentrate on context-aware design. Therefore, the model is able to organize and partition the information, to adapt the layout based on the screen format both horizontally and vertically and to extract style information from other pages in the application.

Context-aware design makes use of several different ICT technologies:

1. **Object-Oriented software Programming** (OOP) allows us to divide and reuse code. The concept is used to solve a wide range of ICT problems such as Artificial intelligence, simulations or graphic design. In our implementation, we apply object-oriented to Cascading Style Sheets (CSS) to give a new dynamic to our style elements.

2. **Design-Patterns** constitute a solid base for building flexible and reusable software. They are one strong output of OOP. They are used in software development as best practices to handle specific scenarios. In our paper, we focus on the JAVA Design Patterns, created to write tree hierarchies of objects. (Hannemann & Kiczales, 2002)

3. **Process Automation** technics serve us to limit the human implication in the development process of the application. In our research, we attempted to adapt a layout to an existing application without the help of a web-designer. Some JAVA design patterns are helpful to produce content without a client application. Finding and defining a good pattern to produce flexible code is mandatory to produce our automation. (Hannemann & Kiczales, 2002)

4. In addition to our theoretical approach, gathered in B. Ballard book on UI design patterns, **UX best practices** are gathered through examples of popular websites. We have analysed a range of newsfeeds in those platforms to produce our card-based newsfeed. (Ballard, 2007; Herring et al., 2009)

5. **Data-driven design** technics have been gathered in the award-winning research paper Webzeitgeist. The core idea behind the research is to apply data-mining technics to a wide range of webpages (103'744) and 12.7 million DOM nodes to analyse their design elements. Webzeitgeist is one starting point for this paper, where we used concept such as data-extraction, and best practices to build our proof of concept. (Kumar et al., 2013)

The approaches studied in this section are used during the whole automation process. In our model, we divide the design in three concrete layers: Information, Layout and Style. In each of those layers, we used a combination of our design automation approaches.

4 Design

In our research, we produce a newsfeed plug-in screen that adapts itself to context:

1. **Style-awareness:** Application it is plugged-in to.

2. **Layout/Device-awareness:** Device or size of browser window it is displayed on.

3. **Information-awareness:** Type and amount of information that needs to be displayed.

The model use different techniques learnt while studying object-oriented design (Composite pattern, media queries, Bootstrap, LESS), UI/UX best practices (Inspiration from most popular way to display newsfeeds, Pinterest vertical handling, Mobile user-experience design patterns) and Design mining (DOM extraction, LESS rendering).

We create a model to plug this screen in multiple application in the future. The model handles a newsfeed production for proof of concept, but the concept of the framework would evolve and will be able to create all kinds of plug-ins for websites.

4.1 Context-aware design framework (CADF) for automated newsfeed plug-ins

The framework tries to automate the design phase of a newsfeed screen by analysing and segmenting the information, adapting to its screen measurement by proposing the best layout possible on every support and by scanning the existing UI of an application to naturally include new screen into existing application.

To define the model we used a combination of UX/UI best practices, Design-mining approach and object-oriented design. Three different layers compose the proposed design model:

A) Information Layer

B) Layout Layer

C) Style Layer

Information of any type must be well processed and sorted to be connected together in an efficient way, the layout of the application must be adapted to the machine it is displayed on. Our screen must contain the same graphical charter than the application it is plugged-in. Pages that the model analyse where already part of the application before the intervention of our plug-in. Those pages are created following a standard coding procedure.

4.1.1 **The three design layers of CADF**

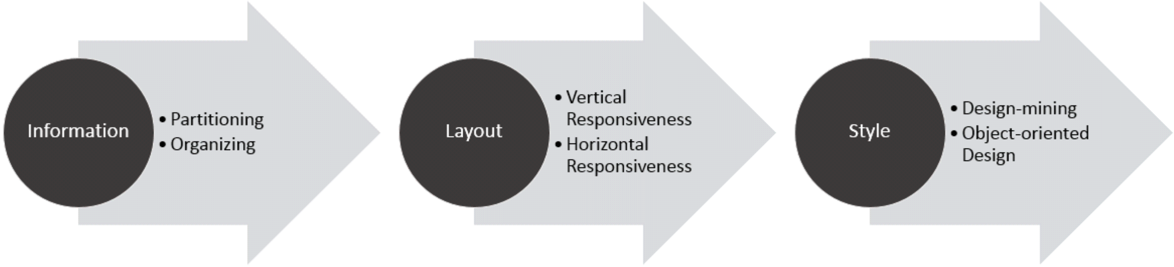


Figure 4 Context-aware design layers and sub processes

By combining those elements, those layers are able to produce a context-aware newsfeed plug-in that adapt its design on the application it is plugged in. We divided the design process into layers and sub-processes to activate them only when a change is needed. In our mHealth case, not all of the design components have to change every time a nurse displays the screen. Styling processes and information organization are handled the first time the screen is plugged in an already-existing application. On the other hand, the Layout Layers have to be adapted every time, as well as the Information partitioning sub-process. In the following sections, we elaborate on the different layers.

4.1.1.1 The Information Layer

Information is a critical part and maybe the most challenging aspect to handle in a context-aware design.

The information layer provides a solution to two different problems:

How do we partition the information/data input to be consistent?

The data partitioning of our framework is design as a Java program.

Object-oriented programming lets us create flexible data models. The Composite pattern for example, used in JAVA programming lets us create sub-model into bigger models. It is particularly useful when it comes in the case of context-aware design because it organizes the information into a hierarchy tree, like XML, JSON or HTML. It is then easier to port data from one of those programming language, transform it into information and export it to another language. (Hannemann & Kiczales, 2002)

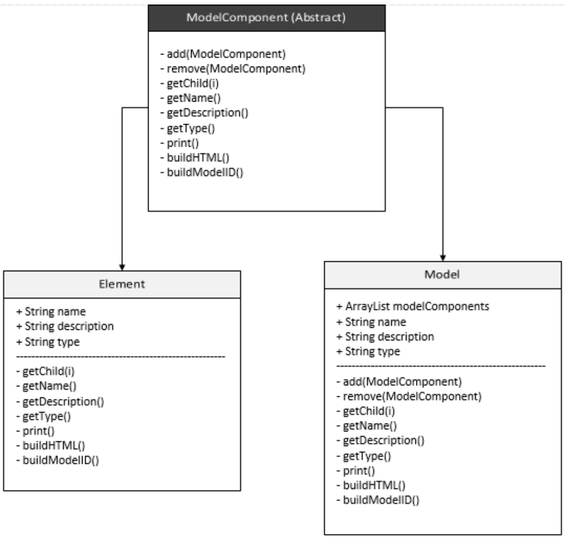


Figure 5 Java Composite Pattern class diagram for CADF

In our Design, we serialize JSON objects into our composite hierarchy. The java program lets us create an HTML file with the information retrieve in the raw JSON file. Our idea behind using a composite pattern is to handle a constant changing input of information into a proper html file. As our goal here is to produce a newsfeed with information cards, we have to go through our JSON objects to identify where the data that should be process are. We try to go as deep as possible to highlight where an information should be handle in our newsfeed. The first step of our partitioning algorithm is to add, as a tree hierarchy, elements to the final model. Composite patterns are created to embed models and elements within bigger models. When we have successfully implement all our models and element into our pattern, we can use our buildHTML() method. The goal of this method is to create html based on information introduced within the algorithm about our elements and models.

As shown in figure 3, each component of the composition have a name, a description and a type. We use the type attributes to add different css classes to our components when they are transformed in HTML. In our htmlsynthaxcon-structor.java class, each type is handled as a switch case structure. It allows us to create objects in a very flexible way and adds defaults design to new type of content.

As a result, the hierarchy is processed in the right order to write a proper HTML file from a tree hierarchy. The core advantage of the Composite pattern is to be able to walk through the class hierarchy and therefore, it perfectly suits our needs in terms of file creation. (Hannemann & Kiczales, 2002)

How do we organize unrelated information in an information flow (news feed, blog post, image gallery)?

Examples and Design Patterns are the approaches we choose to organize and design information. The solution explored in this research was the card - box system, which separate blocks of information in definite cards/boxes. By separating blocks of information visually, we considered that the elements are not related to each other. Figure 3. Presents sample screen captures.

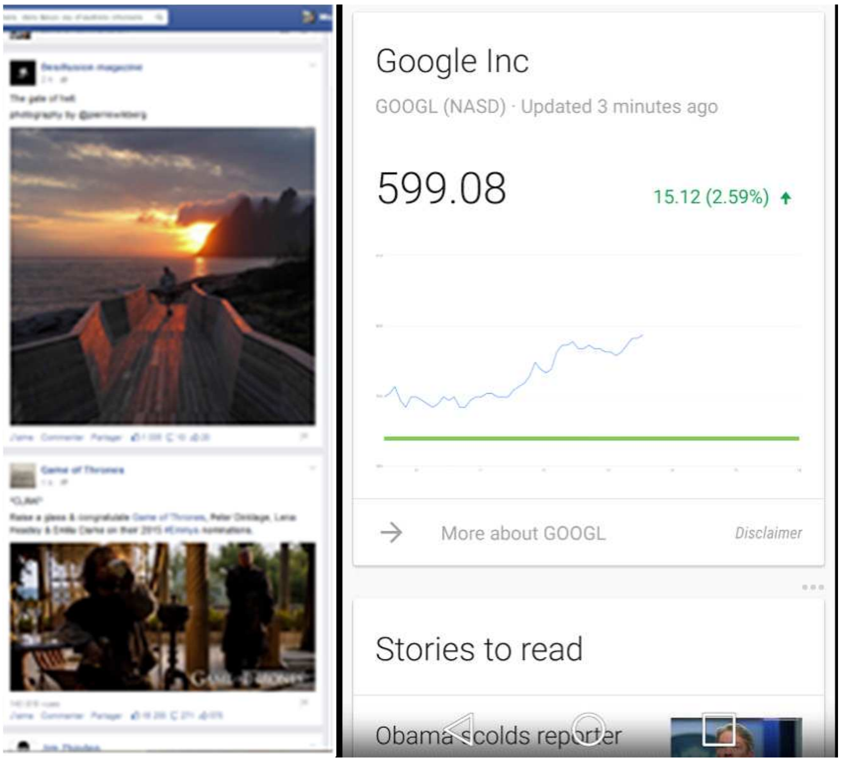


Figure 6 Facebook newsfeed and Google now card layout

This design, widely accepted and used by the market leaders is a good way to visually divide the content. Facebook, Twitter, Google Now, and Instagram, Pinterest and many other major newsfeed producers are using a card-box layout. We create directly a card box design in pure CSS and store it into a file called information.css

4.1.1.2 The Style Layer

After Information handling, we need to adapt the style of our screen to its context. The style context is the application our screen is plugged-in. By dynamically adapting colors and fonts to create a uniformity between our screen and the rest of the application, the design will be able to be added in any solution. Therefore, we need to perform two distinct tasks:

4.1.1.2.1 Style extraction

In Webzeitgeist, they extracted the style of each component of a wide range of websites pages. In this paper, we try to replicate some of the technics they used there to get the graphical charter of our app and adapt our newsfeed to the rest of the application it is plugged in.

By analysing the DOM elements of our application, we are able to define the different colors and fonts, used to design the original application.

Style attributes are stored in browser's cookies. We store a JSON array in the browser to further use when we dynamically style our newsfeed.

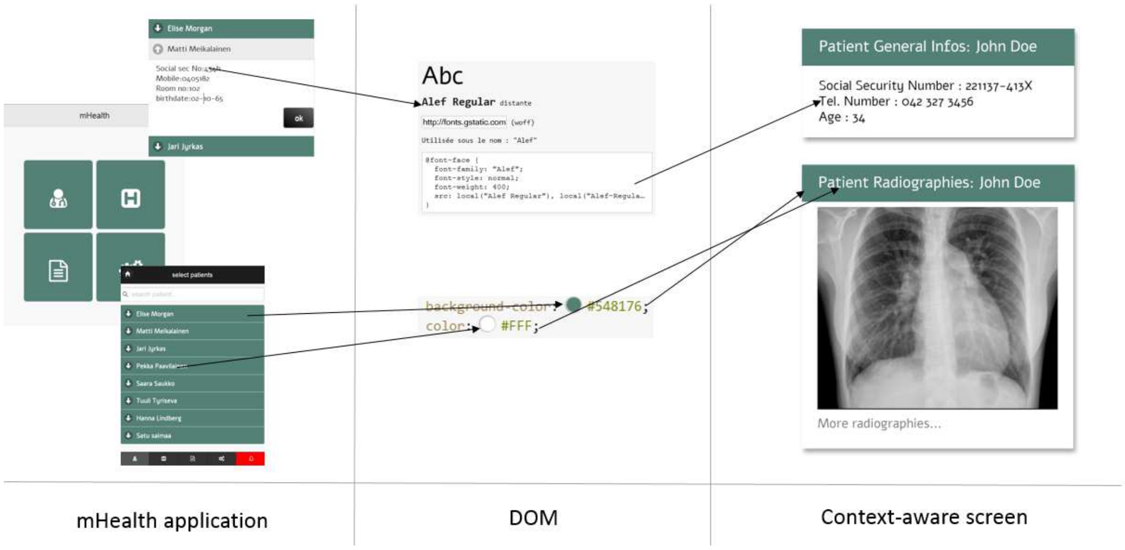
4.1.1.2.2 Dynamic Style Processing

We deserialize the JSON array containing our style elements, stored in browser's cookies. After getting all the style data from the DOM elements, it is crucial to be able to change the style variables of our application. As explained in section V., CSS pre-processor allow us to add variable to stylesheets. By extracting and analysing the main dark and light colors, we are able to integrate smoothly new screens to an application.

We extract DOM style elements via JQuery. We then analyze the color HEX codes of the elements we extracted to define if they are dark or light colors. It is then possible to assign the main dark color to the background of the card's header and the main light color for the card's title. The fonts are sorted by used in titles or paragraph to ensure a uniformity with the rest of the application.

Those variables are pre-processed in our LESS document. When compiled the CSS stylesheet is composed by the rightful colors and fonts, to fit with the rest of the mobile application.

This process allows us to style new screens without having to code anything more. As the style adapts to new context, it would now be possible to add blogs, newsfeeds, slideshows or any kind of popular web pages to an existing application. Figure 7 presents the mHealth DOM extracted style.



Social sec No:456
Mobile:0405182
Room no:102
birthdate:02-10-65

Utilisée sous le nom : "Alef"

```
@font-face {
  font-family: "Alef";
  font-style: normal;
  font-weight: 400;
  src: local("Alef Regular"), local("Alef-Regula…
}
```

```
background-color: #548176;
color: #FFF;
```

Social Security Number : 221137-413X
Tel. Number : 042 327 3456
Age : 34

Figure 7 DOM extraction and context-aware styling in mHealth

4.1.1.2.3 The Layout Layer

With the use of mobile devices in a private and professional use, it has become mandatory to take into consideration user needs to read information on any screen. Therefore, a wide range of web-application uses responsive design and take very well advantage of all the width of a screen, but there is still a lot of screen pixel loss.

Pinterest is a very good example of a company that take any pixel available on screen to display something meaningful to the user. They handle both vertical and horizontal responsiveness. Therefor the use of columns to organize content is a good option to show as many information as possible. In our framework design, we adopt the vertical and horizontal responsiveness. (Pinterest.com, n.d.)

The context is changing at every use as well. However, we do not need to run a specific web service to handle that part as it is integrated directly into our CSS code by dealing with @media queries.

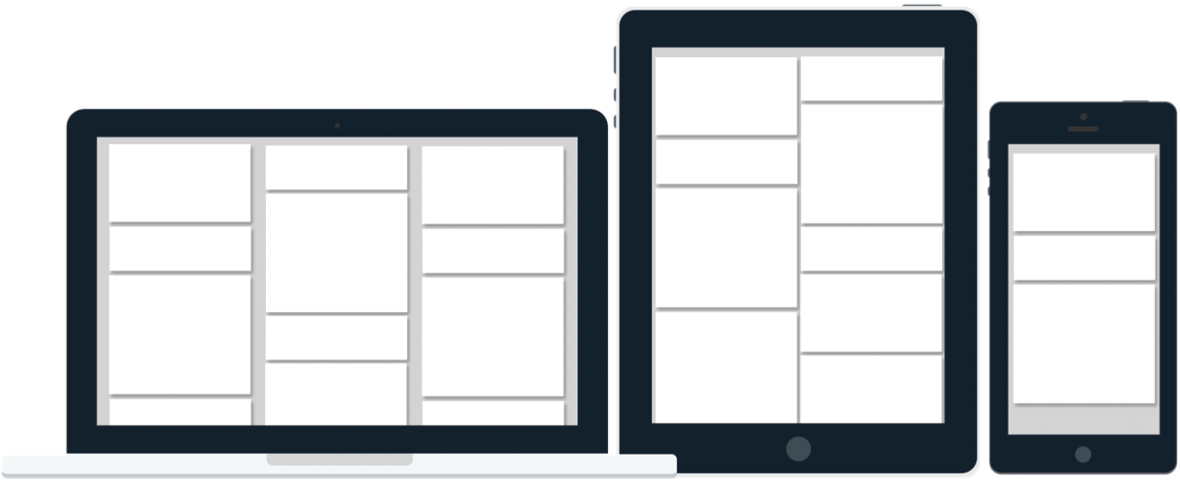
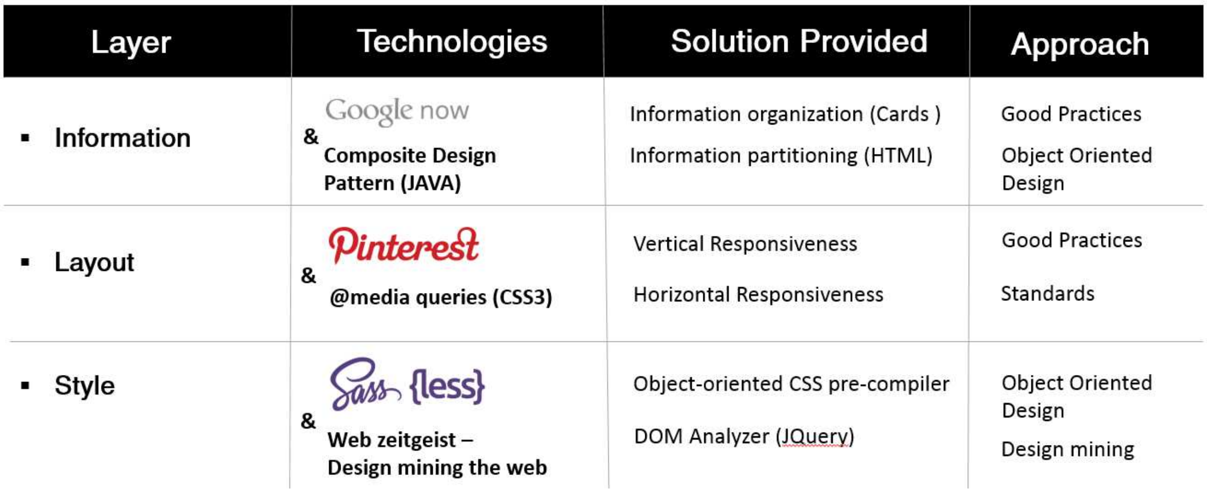


Figure 8 Horizontal and Vertical responsiveness handling in CADF

4.1.2 **CADF workflow**

After designing each sub-process individually to find a solution to our problems, we merge them into a single workflow, called context-aware design framework (CADF) workflow. Our workflow handles all the sub-processes, described in the section 4.1.1, with a combination of technics learn in the different approaches, described in section 3.2.2.1.



Layer	Technologies	Solution Provided	Approach
▪ Information	Google now & **Composite Design Pattern (JAVA)**	Information organization (Cards) Information partitioning (HTML)	Good Practices Object Oriented Design
▪ Layout	Pinterest & **@media queries (CSS3)**	Vertical Responsiveness Horizontal Responsiveness	Good Practices Standards
▪ Style	Sass {less} & **Web zeitgeist – Design mining the web**	Object-oriented CSS pre-compiler DOM Analyzer (JQuery)	Object Oriented Design Design mining

Figure 9 Overview of CADF model

Combining different layers and sub-processes written in different languages is a long process. We need to make sure that our components are working nicely together, in the same development environment (e.g. server compatibility). In some case, we have to add middleware to trigger specific processes in our workflow. Front-end development, in our context-aware and multi-platform case, needs to be coded entirely in HTML5/LESS/CSS/JavaScript to be supported by multiplatform tools that produces IOS, Android or Windows phone applications for mobile devices. Our back-end needs, on the other hand, to use powerful and object-oriented languages. Therefore, we have to connect the two sides with pure HTTP Requests. Those requests can be sent, via Ajax, from the front-end part of our app and process by PHP or Java in our backend. Those language requirements need to be considered before going into a case study implementation, for proof of concept. (Garrett, 2005)

Seven different languages are used to produce our CADF model: (JAVA, JavaScript, HTML, CSS, LESS, PHP and JSON). (Hutchison, 2005; JSON.org, 2006; Shin, Architect, & Microsystems, 2006; W3Schools, 2014; Whatwg, 2011)

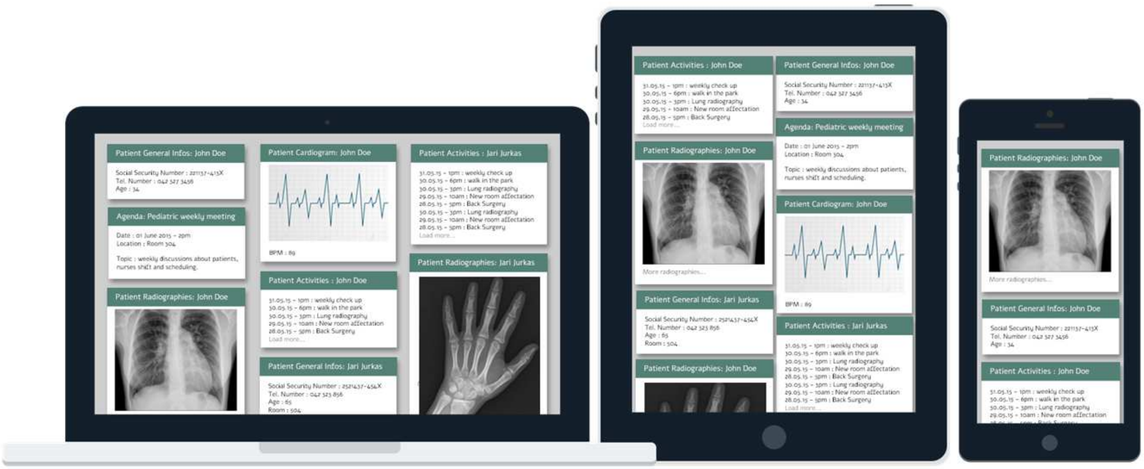
We used a wide range of languages to be as flexible as possible, regarding the context of use of the application. From extracting style elements, cookies storage, AJAX requests to the backend or redirection. JavaScript is handling the front-end work of the application. JAVA is used to create adaptive design patterns and strong object oriented code.

On top of languages use to handle specific sub-processes of the workflow, we use PHP as a middleware language to link backend features, such as our partitioning algorithm, written in JAVA and the front-end of the application, that is written in pure HTML5/CSS3 and JQuery to be supported in native mobile application portage.

We use open standard JSON format to produce our raw data. As the data might be used in other application, our goal here is to find the lightest and most accessible language to read data from.

Combining the various layers of the proposed model is the most critical part of the workflow since we have a wide range of language that should run in a web-server environment and a lot of constraint due to the large amount of concept that mHealth is exploring. Furthermore, the Information layer is very dynamic, especially partitioning the information, which has to be trigger every time the newsfeed need to display information. The card format on the other hand was code ones, inspired by standard web-designs.

The Style layout is handle and process the first time we plug our context-aware newsfeed and extract and process the styling. Figure 10 presents mHealth design automation screencaptures.



Patient General Infos: John Doe

Social Security Number : 221137-413X
Tel. Number : 042 327 3456
Age : 34

Agenda: Pediatric weekly meeting

Date : 01 June 2015 – 2pm
Location : Room 304

Topic : weekly discussions about patients, nurses shift and scheduling.

Patient Radiographies: John Doe

Patient Cardiogram: John Doe

BPM : 89

Patient Activities : John Doe

31.05.15 – 1pm : weekly check up
30.05.15 – 6pm : walk in the park
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation
28.05.15 – 5pm : Back Surgery
Load more...

Patient General Infos: Jari Jurkas

Social Security Number : 2521437-454X
Tel. Number : 042 323 856
Age : 65
Room : 504

Patient Activities : Jari Jurkas

31.05.15 – 1pm : weekly check up
30.05.15 – 6pm : walk in the park
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation
28.05.15 – 5pm : Back Surgery
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation
28.05.15 – 5pm : Back Surgery
Load more...

Patient Radiographies: Jari Jurkas

Patient Activities : John Doe

31.05.15 – 1pm : weekly check up
30.05.15 – 6pm : walk in the park
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation
28.05.15 – 5pm : Back Surgery
Load more...

Patient Radiographies: John Doe

More radiographies...

Patient General Infos: Jari Jurkas

Social Security Number : 2521437-454X
Tel. Number : 042 323 856
Age : 65
Room : 504

Patient Radiographies: John Doe

Patient General Infos: John Doe

Social Security Number : 221137-413X
Tel. Number : 042 327 3456
Age : 34

Agenda: Pediatric weekly meeting

Date : 01 June 2015 – 2pm
Location : Room 304

Topic : weekly discussions about patients, nurses shift and scheduling.

Patient Cardiogram: John Doe

BPM : 89

Patient Activities : Jari Jurkas

31.05.15 – 1pm : weekly check up
30.05.15 – 6pm : walk in the park
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation
28.05.15 – 5pm : Back Surgery
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation
28.05.15 – 5pm : Back Surgery
Load more...

Patient Radiographies: John Doe

More radiographies...

Patient General Infos: John Doe

Social Security Number : 221137-413X
Tel. Number : 042 327 3456
Age : 34

Patient Activities : John Doe

31.05.15 – 1pm : weekly check up
30.05.15 – 6pm : walk in the park
30.05.15 – 3pm : Lung radiography
29.05.15 – 10am : New room affectation

Figure 10 Context-aware screen after design automation following CADF model

5 Implementation

We applied our CADF framework on mHealth project to produce a newsfeed screen that will display meaningful information to the user based on his/her context (User's behaviours- Localisation- and Design-awareness).

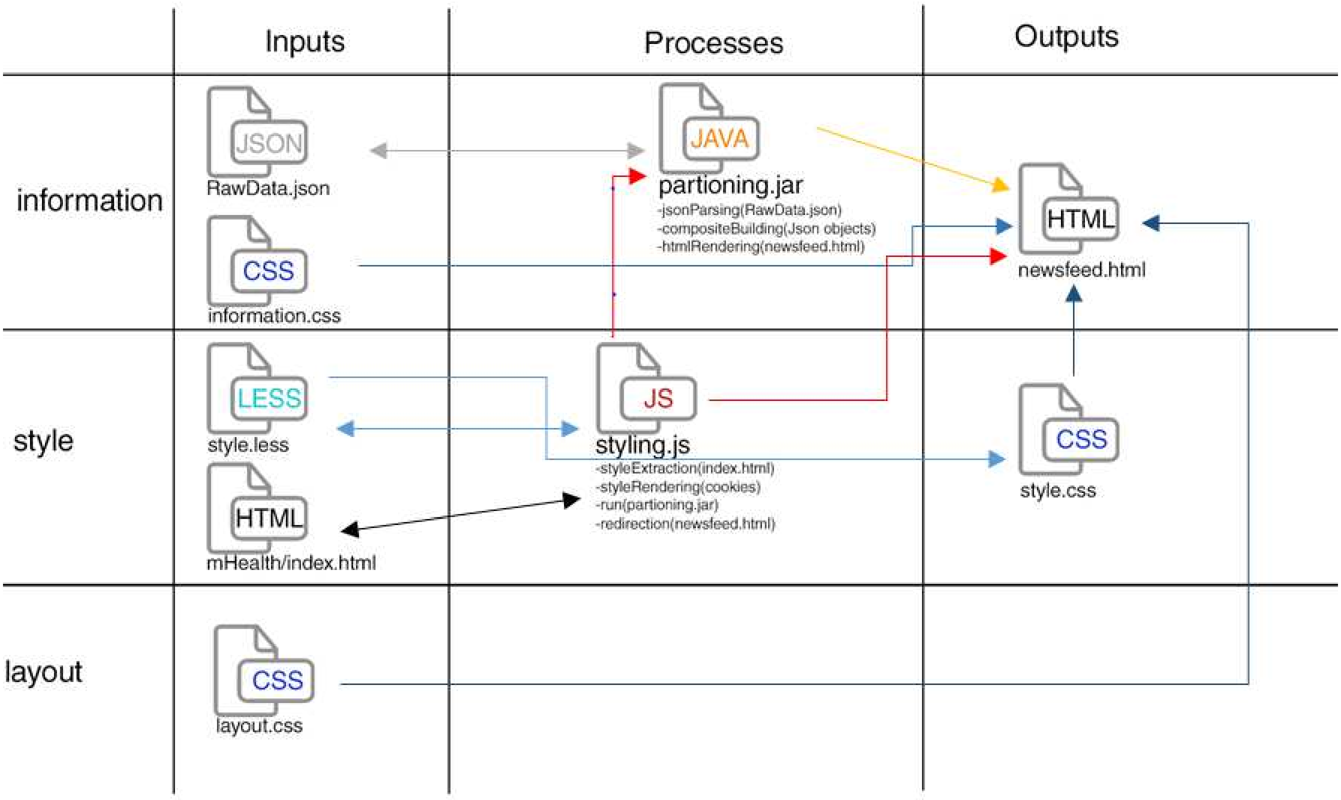


Figure 11 CADF workflow in mHealth

5.1 Inputs in mHealth

Following our three layers principle, inputs needed in CADF framework are:

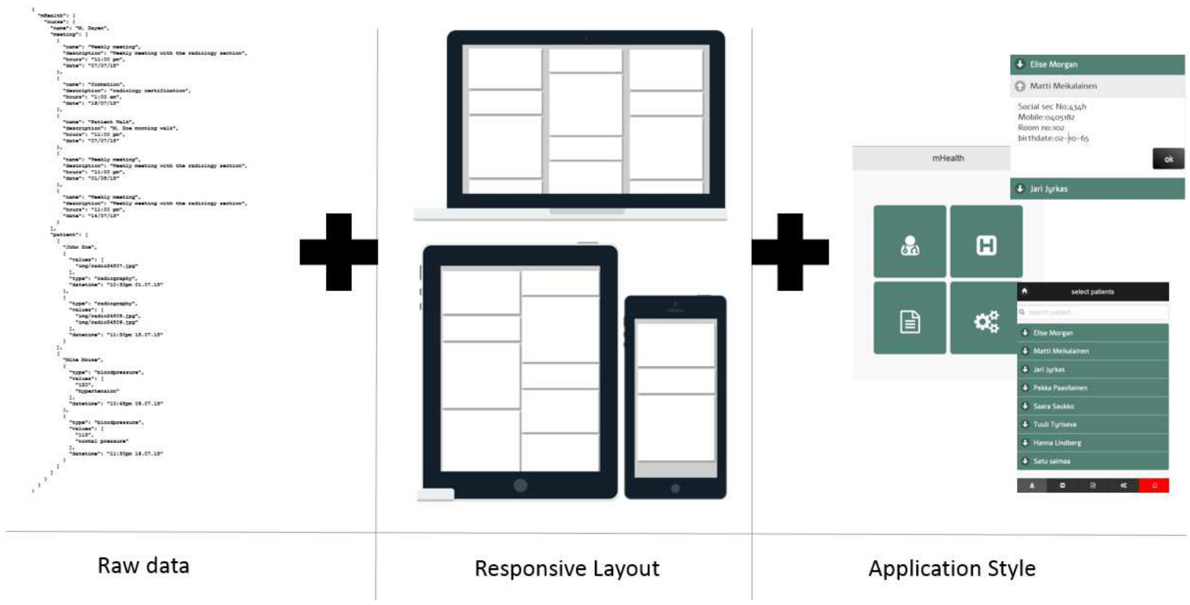
Information: Raw data is given by a JSON file, chosen randomly between three files we created to simulate a user's behaviour and location-awareness, that are part of related studies on mHealth project but not explored in this paper. We also use information.css as a style reference for our information card layout.

Layout: The layout responsiveness is handled by our layout.css file.

Style: We use the style provided in the home screen of mHealth project as a starting point for our style extraction and rendering (index.html). As a proof of concept, we use the style element contained in the DOM element #extractedStyle.

Figure 12 #extractedStyle element: our style starting point

Now that we have all the sources that we need to construct automatically a design, we can process everything in our 3 processes files.



```
{
  "mHealth": {
    "nurse": {
      "name": "M. Zayen",
      "meeting": [
        {
          "name": "Weekly meeting",
          "description": "Weekly meeting with the radiology section",
          "hours": "11:00 pm",
          "date": "07/07/15"
        },
        {
          "name": "formation",
          "description": "radiology certification",
          "hours": "1:00 am",
          "date": "18/07/15"
        },
        {
          "name": "Patient Walk",
          "description": "M. Doe morning walk",
          "hours": "11:00 pm",
          "date": "07/07/15"
        },
        {
          "name": "Weekly meeting",
          "description": "Weekly meeting with the radiology section",
          "hours": "11:00 pm",
          "date": "21/06/15"
        },
        {
          "name": "Weekly meeting",
          "description": "Weekly meeting with the radiology section",
          "hours": "11:00 pm",
          "date": "14/07/15"
        }
      ],
      "patient": [
        [
          "John Doe",
          {
            "values": [
              "img/radio24507.jpg"
            ],
            "type": "radiography",
            "datetime": "10:30pm 01.07.15"
          },
          {
            "type": "radiography",
            "values": [
              "img/radio24508.jpg",
              "img/radio24509.jpg"
            ],
            "datetime": "11:30pm 15.07.15"
          }
        ],
        [
          "Mike House",
          {
            "type": "bloodpressure",
            "values": [
              "120",
              "hypertension"
            ],
            "datetime": "10:45pm 08.07.15"
          },
          {
            "type": "bloodpressure",
            "values": [
              "115",
              "normal pressure"
            ],
            "datetime": "11:30pm 19.07.15"
          }
        ]
      ]
    }
  }
}
```

Figure 13 Inputs in mHealth

5.2 Processes in mHealth:

Style: Our JavaScript file styling.js is the first process to be activated to pass from our source screen (index.html) to our context-aware newsfeed plug-in.

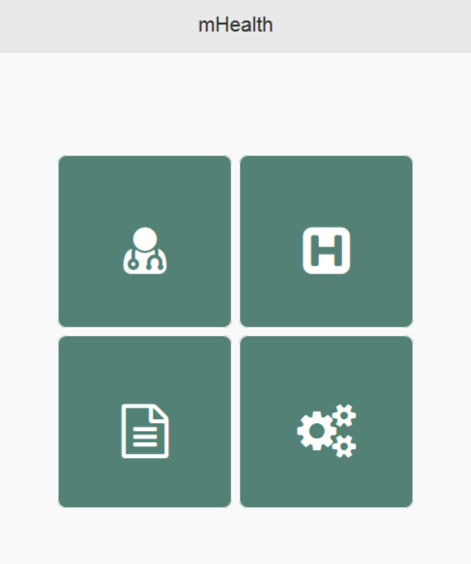
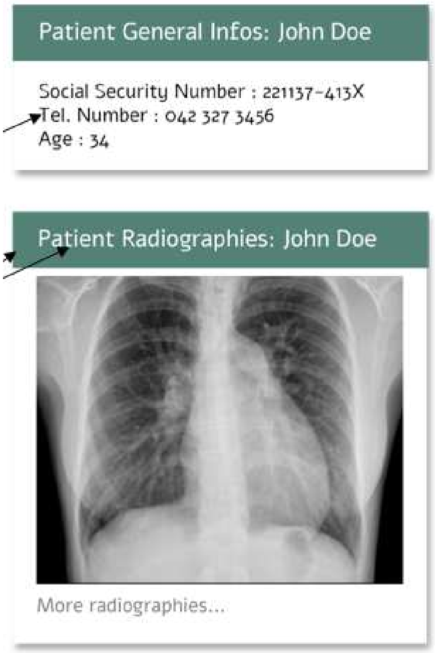


Figure 14 mHealth home screen

The file, as explained in the section 4.1.1.2, will extract style, store it in browser cookies and then trigger the java file and redirect us to our newsfeed. The Styling.js file is the main script of the design workflow. The latest file is triggered by clicking on the report button (bottom-left) of mHealth home screen.


Patient General Infos: John Doe

Social Security Number : 221137-413X
Tel. Number : 042 327 3456
Age : 34

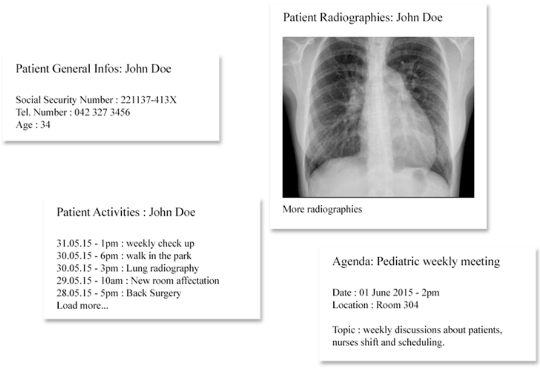
Patient Radiographies: John Doe

More radiographies...

Figure 15 Card that have process the style of mHealth

However, for the implementation of the model with mHealth, we encountered a few problems extracting style from every browser, e.g., Internet Explorer uses a proper method to render styles. (2) The extracted styles have to be changed in our LESS file. But for security reasons, overwriting files in JavaScript is difficult. Therefore, we choose to store the DOM elements in browser cookies and render the LESS file when we displayed our newsfeed. (3) We trigger our Java executable to dynamically create our newsfeed. (4) We finally redirect the user to newsfeed.html.

Information: Our Java partitioning.jar contains three main methods: (1) A JSON parser to extract the raw data from our file. (2) We create a hierarchy model with this data, by using a composite pattern, we create models and elements that are imbricated together. (3) A HTML page is produced as a hierarchy and then we render our composition into newsfeed.html. However, in our context, the composite pattern has some limitations. The result in mHealth is showed by dividing the information by sitemap situation in the rest of the application. It allowed us to display multiple information elements, which are not related to each other without confusing the users. The composite pattern partitions organize the information directly into a HTML5 file. The file is then styled by a simple card format, without any font or colour styling. We define a standard width of the cards and the height adapt to the object display. Additionally, the objects of the mHealth personal newsfeed is created. Hence, we can now display them into a layout that try to minimize the pixel loss and style the cards to fit the global design of mHealth application. Figure 16 displays sample of screen captures.



Patient General Infos: John Doe

Social Security Number : 221137-413X
Tel. Number : 042 327 3456
Age : 34

Patient Radiographies: John Doe

More radiographies

Patient Activities : John Doe

31.05.15 - 1pm : weekly check up
30.05.15 - 6pm : walk in the park
30.05.15 - 3pm : Lung radiography
29.05.15 - 10am : New room affectation
28.05.15 - 5pm : Back Surgery
Load more...

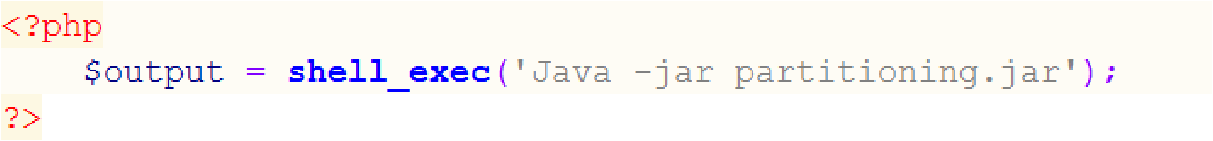
Agenda: Pediatric weekly meeting

Date : 01 June 2015 - 2pm
Location : Room 304

Topic : weekly discussions about patients, nurses shift and scheduling.

Figure 16 Information partitioning and organization with a card layout in mHealth

Middleware: As JavaScript does not allow file execution, we had to create a PHP middleware to run our Java .JAR file. The PHP script is triggered by an Ajax GET request. PHP allows us to use exec("java –jar myfile.jar") or shell-exec() functions to run our file.



```
<?php
    $output = shell_exec('Java -jar partitioning.jar');
?>
```

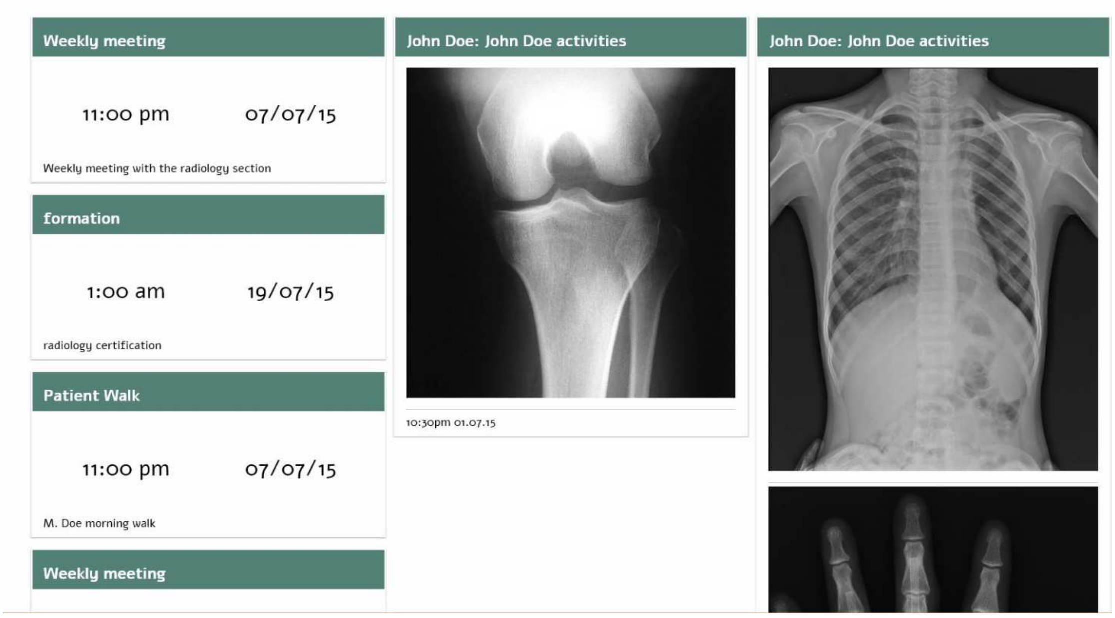
Figure 17 Our PHP middleware

5.3 Outputs in mHealth

Finally, CADF proof of concept's outputs were produced into newsfeed.html. Once we are redirected to our newsfeed.html page, we take the information stored in the browser cookies and apply them dynamically to our LESS file. Once the data are added into LESS variables, the LESS.js file is processing the file and render it into a .css file.

Directly linked with information.css and layout.css, our newsfeed is now display with a minimal pixel loss, both vertically and horizontally, with meaning full data provided after user behavior studies and styled as the rest of mHealth graphical interface.

The layers combined, we can observe the design and analyses the output by validating context-awareness and usability.



Weekly meeting
11:00 pm 07/07/15
Weekly meeting with the radiology section

formation
1:00 am 19/07/15
radiology certification

Patient Walk
11:00 pm 07/07/15
M. Doe morning walk

Weekly meeting

John Doe: John Doe activities
10:30pm 01.07.15

John Doe: John Doe activities

Figure 18 mHealth newsfeed screen

5.4 Difficulties encounter during the implementation

The biggest implementation issue of our model was to use the best possible language for each component of the framework and still make it work nicely together in one workflow. We encountered issues to run our Java-based content in a web-environment. To solve it, we use a PHP middleware to connect the front-end of the application with our java-based algorithm, in the back-end of the application.

We weren't able to dynamically create levels of content. We had to create a specific model to handle Meetings and a specific model to handle Patient Activities. In a future study, we will dig further in Design Pattern mechanics to find a better way to dynamically organize content.

We also encounter issues to deploy our solution in the main server that runs mHealth prototype because we are storing some C# back-end services. Those services can only be hosted in a windows server environment. On the other hand, our solution makes use of a PHP middleware that needs to be hosted in an Apache Environment. With a bigger timeframe of development, we could host the front-end in one environment and those back-end features in separate servers.

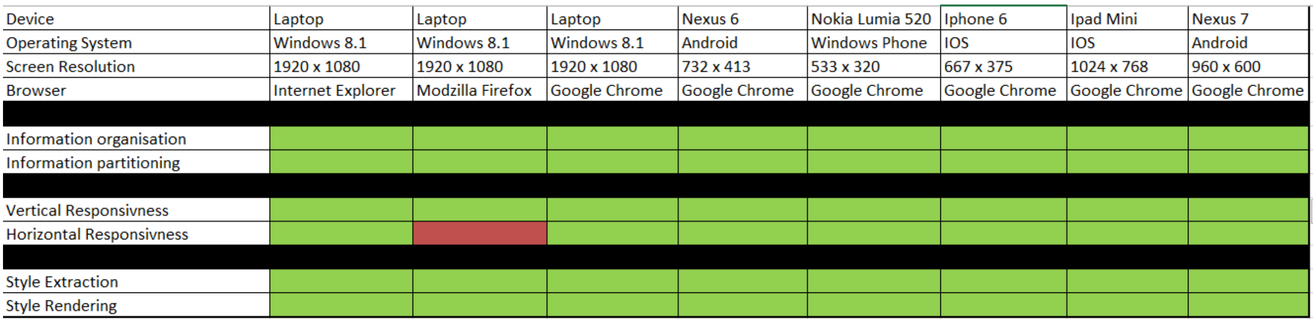
6 Model Assessment & Analyze of the results

6.1 Assessment of the model

We designed a solution to create a semi-automated and dynamic design for a newsfeed screen. We implemented it within a healthcare project successfully by following CADF.

Screen and Device awareness and responsiveness are key elements to assess. We have run tests on 3 different browsers on laptops (Internet Explorer, Mozilla Firefox and Google Chrome), and test our layout on 5 mobile devices from different providers (Google, Apple, and Nokia). We have run our test on a smartphone and a tablet from Google, the Nexus 6(smartphone) and Nexus 7(tablet), both running Android and Google chrome browser. We have also run our compatibility test with an IPhone 6(smartphone) and a iPad mini(tablet). Nokia Lumia 520 was also included in the device responsiveness test. As we didn't had the whole range of devices for our evaluation, mobile devices where emulated within the developer console of Google Chrome browser. (Apple, n.d.-a, n.d.-b; Asus, n.d.; Microsoft, n.d.; Motorola, n.d.)

To assess our research on all those device we individually tested all of the sub-processes. Every device and web-browser is running the entire workflow and we assess: (1) the information organization, (2) the information partitioning, (3) the vertical responsiveness, (4) the horizontal responsiveness, (5) the style extraction and (6) the style rendering. Every process is assessed as working (green tile) or not working (red tile).



Device	Laptop	Laptop	Laptop	Nexus 6	Nokia Lumia 520	Iphone 6	Ipad Mini	Nexus 7
Operating System	Windows 8.1	Windows 8.1	Windows 8.1	Android	Windows Phone	IOS	IOS	Android
Screen Resolution	1920 x 1080	1920 x 1080	1920 x 1080	732 x 413	533 x 320	667 x 375	1024 x 768	960 x 600
Browser	Internet Explorer	Modzilla Firefox	Google Chrome	Google Chrome	Google Chrome	Google Chrome	Google Chrome	Google Chrome
Information organisation	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Information partitioning	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Vertical Responsivness	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Horizontal Responsivness	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩
Style Extraction	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Style Rendering	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Figure 19 CADF model implementation assessment in mHealth

On 48 sub-processes tested in a given context, we can see that 47 of them are working as they were expected to. As a wish of high compatibility and device-awareness, we can confirm that CADF model is working on 98% of the devices types.

We encounter a horizontal compatibility problem on Mozilla Firefox. Web browser are not reacting exactly similar to all css attributes. To be compatible with the latest versions of Mozilla we will have to insert specific handling for this browser in our media queries, in our layout.css file.

We also assess our java algorithm by proposing different inputs to our program. We have created two JSON raw data files, each of them are composed by a random number of patient activities and nurses meetings. We are focused on the quantity of inputs that the algorithm is supposed to process. Our two files are processed correctly.

This research paper is the result of the end of a year-long development cycle and presents the final context-aware mobile prototype that mHealth aims to be in the future. The development workflow is completed by the combination of user behavior predictions, location detection and design context-awareness to produce our final newsfeed for any hospital workers. This paper does not use data provided by the other two related research areas (user-behavior prediction and location awareness). The next step will be to link our three distinct studies together to assess the whole mHealth prototype for usability testing in a work environment.

The assessment of the mHealth prototype will require one-on-one interviews with hospital workers that will test contextual laddering. It is used in UX evaluation to better understand the reasons why certain attributes are more important to a single customer and evaluating how well a model is working. The process is long and requires more than an hour of interview for each stakeholder. Those interviews were not done in the in the allotted time-frame. (Vanden Abeele & Zaman, 2009)

6.2 Analysis of the results

The first objective of this paper was to study different approaches to design automation. After reading the state of art of the UI/UX best practices, Design mining and Object oriented design, we have concluded to merge the three approaches to create a suitable model that will fit in this global study of healthcare context-awareness. Furthermore, by taking component of each approach, we were able to cut short the process and partially automate it. The solution needs improvements: such as a better information sorting and webpage scanning to create meaningful machine learning if we want a fully intelligent and evolving framework that will design entire websites by itself.

The combination of approaches will be a good fit to create a bigger framework, with more effort put into design-mining and machine learning. By taking the design mining and extractions methods of Webzeitgeist study, we would be able to produce a sufficient training dataset to produce designs by computer. Going in this direction can produce very surprising outputs by machine learning and it would be worth it to explore the domain, within a proper timeframe.

The second objective of this paper was the production of a newsfeed plug-in for applications. In our mHealth project, our plugin is fully integrated and fits the design framework that we created. The model is assessed in the section 6.1 and 98% of the sub-processes in a given context tested are working as intended.

The web experience evolves and templates are not sufficient enough to produce a strong user experience to customers. As the number of services available online is still growing, it is crucial to find new way to interconnect those services. As corporate websites should include no more than what their specific customer are looking for and should not make concessions on features because they are too long to develop or not included in a template, our solution comes to use of a more modular web experience. By adding more resources to an existing web-product at any time, we allow businesses to conduct their marketing researches and fulfil the needs of their customers faster than ever. With more modularity, a website could evolve as fast as its business counterpart.

6.3 **Improvements**

With more time, it would be possible to create new modules for already existing websites. This new approach is very interesting for the web development industry because it will be possible to add more features to an existing web-site. We can imagine that startups specialized in some web component will provide those kinds of plug-ins in the future. For example, we would be able to add a blog that will automatically be designed like the rest of our application. This blog can be provided by Blogger for example and therefore integrate all their back-end management features.

The process AS-IS will need more uniformity and some programming languages should not be used in the process. PHP for example, was used to connect JAVA and AJAX/JQuery together. Those two languages can work together, using JAVA MVC framework. The reason we did not use such technologies was to reduce the development timeframe and to be able to produce our proof of concept in the given time. JAVA MVC requires the setup of a Linux Apache server that can run specifically java, using Apache TOMCAT hosting.

An improvement that can be made is the production of a well order JSON file containing all the information that needs to be display, shared from the backend via web API technologies and process by AJAX on the front-end of our application. In our current workflow, we use a composite pattern to directly create a static webpage to display our content.

The backend of the application can use other technics in further development. If we decide to integrate Machine Learning in our process, we could use KNIME technologies, based on Java. KNIME is a machine learning open-source platform that allow users to create and compare machine-learning and data-mining algorithm to add artificial intelligence on top of our application. We use it to predict user behavior in mHealth. The software is built on top of Eclipse IDE. Those technologies should be implemented in dedicated servers, so we may have to change our back-end architecture. Nevertheless, the front-end of our application will still run properly in a multiplatform environment. (KNIME.COM AG, n.d.)

As an improvement, we could as well sort all the cards created before writing our file. By processing the date of each cards as an integer (e. g. 01092015) we could use a Quicksort algorithm. The sorting algorithm will take too much time to implement and is not critical to the demonstration of our model. It will be implemented in further studies on the CADF model. (Java, n.d.)

7 Conclusion

This was a proof of the concept that our proposed model is functioning as expected. Therefore, the full implementation of the automation by machines is planned to be implemented in mHealth application. Hence, in this paper, we have explored a solution to create a modular web development framework. By orienting our research to cut short development time, with the objective of automating the whole design process, we have discovered an interesting approach to a modular web. Therefore, in this paper, we have explored the fundamentals of design automation and not the actual implementation. In further research, we have plans to build machine learning on top of design-mining datasets to predict and create standardized web pages. We will concentrate our efforts to create other plug-ins, such as blogs for example. Hence, it will enables to create a dynamic webpage and plug it into another application. Additionally, different alternatives will be considered in the future to organize information in the plug-ins because Java executable are not supported by multiplatform technologies, such as PhoneGap.

By taking an already existing website and adding new features on it, that adapt themselves to the environment they are plugged-in, we can create new web frameworks that are more flexible. The main advantage of the solution proposed by this paper is the modularity of the content. From a business perspective, not every customer has the same needs and preferences. Examples are strongly adopted by people to choose wisely which sort of layout they want to include on their websites. By allowing developers to create and sell self-made plug-ins for the web, we will see more personalization in websites. Made by professionals, those featured plug-ins will maintain UX and engineering standards. We see great potential in this new approach, in comparison to template-based websites. Rather than being obliged to choose the perfect template to suit customer needs, this framework will allow the choice of a wide range of adaptive tools that fulfill their business needs. Economically, it also make sense to buy components you really need, with the quality of service you would need. Modularity is a good approach to leverage the quantity and quality of effort, IT professionals and business owners must put into the creation of a website.

8 References

Abras, C., Maloney-krichmar, D., & Preece, J. (2004). User-Centered Design. *Design*, *37*(4), 1–14. http://doi.org/10.3233/WOR-2010-1109

Adobe. (2013). PhoneGap Documentation. Retrieved from http://docs.phonegap.com/en/3.1.0/guide_overview_index.md.html#Overview

Apple. (n.d.-a). Apple - iPad mini 3 - Technical Specifications. Retrieved from http://www.apple.com/ipad-mini-3/specs/

Apple. (n.d.-b). Apple - iPhone 6 - Technical Specifications. Retrieved from http://www.apple.com/iphone-6/specs/

Asus. (n.d.). Nexus 7 specifications. Retrieved from https://www.asus.com/Tablets/Nexus_7/specifications/

Awwwards.com. (n.d.). Awwwards - Website Awards - Best Web Design Trends.

Ballard, B. (2007). Mobile User Interface Design Patterns. In *Designing the Mobile User Experience* (p. 260). John Wiley & Sons. Retrieved from https://books.google.ch/books?id=-RWG1P0EXPcC&printsec=frontcover&dq=Designing+the+Mobile+User+experience&hl=fr&sa=X&ved=0CCcQ6AEwAGoVChMIgczTkPPBxwIVF0aICh0GAgF5#v=onepage&q=Designing the Mobile User experience&f=false

Benyó, B., Vilmos, A., Kovacs, K., & Kutor, L. (2007). The design of NFC based applications. In *INES 2007 - 11th International Conference on Intelligent Engineering Systems, Proceedings* (pp. 277–280). http://doi.org/10.1109/INES.2007.4283711

Dirin, A., & Nieminen, M. (2015). mLUX□ :Usability and User experience development framework for m-learning. *iJIM, International Journal of Interactive Mobile Technologies.*

Dirin, M., Dirin, A., & Laine, T. H. (2015). User-Centered Design of a Context-Aware Nurse Assistant (CANA) at Finnish Elderly Houses. In *The 9th International Conference on Ubiquitous Information Management and Communication*. The Mulia, Bali, Indonesia.

Eppstein, C. M. (n.d.). Compass Home | Compass Documentation. Retrieved from http://compass-style.org/

Fling, B. (2009). Designing for context. In *Mobile Design and Development□ : Practical concepts and techniques for creating mobile sites and web apps* (p. 336). O'Reilly Media, Inc. Retrieved from https://books.google.ch/books?id=LyMeulBTkH0C&printsec=frontcover&dq=mobile+design+and+development&hl=fr&sa=X&authuser=1#v=onepage&q=mobile design and development&f=false

Foster, I., Kesselman, C., & Kesselman, I. F. C. (2004). The Grid.

Garamszegi, L. Z. (2011). Information-theoretic approaches to statistical analysis in behavioural ecology: An introduction. *Behavioral Ecology and Sociobiology*. http://doi.org/10.1007/s00265-010-1028-7

Garrett, J. (2005). Ajax: A new approach to web applications, 1–5. Retrieved from https://courses.cs.washington.edu/courses/cse490h/07sp/readings/ajax_adaptive_path.pdf

Gevarter, W. B. (1987). INTRODUCTION TO ARTIFICIAL INTELLIGENCE. *Chemical Engineering Progress*, *83*(9), 21–37. http://doi.org/10.2207/qjjws1943.57.490

Google. (n.d.). Web Authoring Statistics - Webmasters — Google Developers.

Hannemann, J., & Kiczales, G. (2002). Design pattern implementation in Java and aspectJ. *ACM SIGPLAN Notices*, *37*(11), 161. http://doi.org/10.1145/583854.582436

Herring, S. R., Chang, C. C., Krantzler, J., & Bailey, B. P. (2009). Getting inspired!: understanding how and why examples are used in creative deHerring, S.R., C.C. Chang, J. Krantzler, and B.P. Bailey. 2009. "Getting inspired!: understanding how and why examples are used in creative design practice."P. 87–96 in Proceedin. *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, 87–96. Retrieved from http://portal.acm.org/citation.cfm?id=1518701.1518717

Hutchison, B. (2005). LESS Getting Started.

Java. (n.d.). Quicksort - Wikipedia, the free encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Quicksort

JSON.org. (2006). *The application/json Media Type for JavaScript Object Notation (JSON).*

KNIME.COM AG. (n.d.). KNIME _ Documentation. Retrieved from http://tech.knime.org/documentation

Kumar, R., Satyanarayan, A., & Torres, C. (2013). Webzeitgeist: Design mining the web. *Proceedings of the …*, 3083–3092. http://doi.org/10.1145/2470654.2466420

Laine, M., Shestakov, D., Litvinova, E., & Vuorimaa, P. (2011). Toward unified web application development. *IT Professional*, *13*(5), 30–36. http://doi.org/10.1109/MITP.2011.55

Lubart, T. (2005). How can computers be partners in the creative process: Classification and commentary on the Special Issue. *International Journal of Human Computer Studies*, *63*(4-5 SPEC. ISS.), 365–369. http://doi.org/10.1016/j.ijhcs.2005.04.002

Marcotte, E. (2010). Responsive Web Design. *Internetdocument*, (306), 10. http://doi.org/10.11635/2319-9954/1/1/18

Microsoft. (n.d.). Nokia Lumia 520 - Specifications. Retrieved from http://www.microsoft.com/en-us/mobile/phone/lumia520/specifications/

Motorola. (n.d.). Nexus 6 Specifications.

Oriented, O., Programming, O. O., & Oo, M. (2001). Object Oriented Programming Concepts 3.1. *ACM Sigplan Notices*, *21*(10), 33–49. http://doi.org/10.1145/323648.323751

Pinterest.com. (n.d.). Pinterest. Retrieved from https://www.pinterest.com/

Pttrns.com. (n.d.). Pttrns _ Mobile design patterns, resources and inspiration.

Shin, S., Architect, J. T., & Microsystems, S. (2006). Introduction to JSON (JavaScript Object Notation). *Technology*.

Talton, J. O., Ahmad, S., & Klemmer, S. R. (n.d.). Data-driven Web Design.

Twitter. (2014). Bootstrap 3. Retrieved from http://getbootstrap.com/

Vanden Abeele, V., & Zaman, B. (2009). Laddering the User Experience! *User Experience Methods, Interact 2009,* (August). Retrieved from http://www.kuleuven.be/facdep/social/com/mediac/cuo/admin/upload/Laddering the User Experience.pdf

W3Schools. (2014). JavaScript HTML DOM. Retrieved from http://www.w3schools.com/js/js_htmldom.asp

Whatwg. (2011). HTML Standard. Retrieved from http://www.whatwg.org/specs/web-apps/current-work/

Wikipedia. (n.d.). Sass (stylesheet language) - Wikipedia, the free encyclopedia.

Yang, Q., Pan, S. J., Zheng, V. W., Kashima, H., Suzuki, S., Hido, S., … Chen, Y. (2008). Estimating location using Wi-Fi. *IEEE Intelligent Systems*, *23*(1), 8–9. http://doi.org/10.1109/MIS.2008.4