Requirements of the interaction design pattern languages for the web development: A pattern language delivery platform
Abstract

**Purpose** – Previous studies of pattern languages have been focused HCI pattern languages or presented them in the context web development before advent of touchscreen mobile devices. This study looks at potential and evolvement of pattern languages in general and then evaluates the effects and challenges that modern Web environment brings to development of pattern languages for the web development.

**Methodology** – This research was conducted as design science research by following the process of recognized design science research methodology. Experiment including qualitative survey was conducted in evaluation phase. Narrative literature review was conducted summarizing the most relevant and highest quality knowledge available to answer the research questions as part of the background research.

**Findings** – This research points to that pattern languages have untapped potential in web technology based user interfaces. This study didn't yet provide definitive answer to how to untap this potential, but instead provided set of requirements to build upon.

**Research limitations** – Study focuses on only developing user interface patterns as front-end elements that can be paired with desired backend implementations. Artifacts developed in this study should also be evaluated in larger qualitative research in order to better evaluate the impact.

**Practical implications** – Delivery platform for pattern language was developed and documented as artifact. Basic pattern language was also developed and documented to enable evaluation of the delivery platform. Insights of this paper aims to provide basis for the building pattern languages for the web development. Value of the paper comes also from discussing the role of recent tools and techniques that have been recently adopted in the web development or are just being developed.

**Keywords**
user interface, design, patterns, pattern language, web development, web design
Contents

Abstract........................................................................................................................................... 2
Contents............................................................................................................................................... 3
1. Introduction................................................................................................................................... 5
2. Background................................................................................................................................... 7
   2.1 User interface design patterns................................................................................................. 7
      2.1.1 Definition of user interface design patterns ..................................................................... 7
      2.1.2 History of usage of patterns.............................................................................................. 7
      2.1.3 Tools similar to patterns...................................................................................................... 8
      2.1.4 Benefits of user interface design patterns.......................................................................... 9
      2.1.5 HCI pattern language definition......................................................................................... 9
      2.1.6 Forming design process with pattern languages................................................................. 10
      2.1.7 Utilizing user experience knowledge with patterns............................................................ 11
      2.1.8 Pattern semantics and representation.................................................................................. 12
      2.1.9 Issues with development of pattern languages................................................................. 13
   2.2 User interface pattern challenges in the Web environment....................................................... 14
      2.2.1 Benefits and issues of responsive design......................................................................... 14
      2.2.2 Recent approaches in pattern languages for the Web....................................................... 15
      2.2.3 Issues with adoption of tools in web development.............................................................. 15
      2.2.4 Challenges in web pattern interpretation.......................................................................... 16
      2.2.5 Patterns as a tool for solving accessibility issues................................................................. 17
      2.2.6 Patterns as a tool for prototyping for the web.................................................................... 18
      2.2.7 Patterns as a tool for persuasion for the web..................................................................... 19
3. Research Method.......................................................................................................................... 21
   3.1 Design Science and Information System Research.................................................................. 21
   3.2 Adopted Design Science Research Methodology process....................................................... 23
      3.2.1 Methodology.................................................................................................................... 23
      3.2.2 Implementation of DSRM methodology........................................................................... 24
4. The Software Artifact.................................................................................................................. 26
   4.1 Motivation and contribution..................................................................................................... 26
   4.2 Requirement Specification and objectives................................................................................ 26
   4.3 User Interface Design of the delivery system......................................................................... 30
      4.3.1 Home page....................................................................................................................... 31
      4.3.2 Site types view.................................................................................................................. 32
      4.3.3 Browse layouts view......................................................................................................... 33
      4.3.4 Customize layout view...................................................................................................... 34
      4.3.5 Pattern view...................................................................................................................... 35
      4.3.6 Admin views..................................................................................................................... 36
   4.4 Software Architecture.............................................................................................................. 37
      4.4.1 Underlying technologies.................................................................................................. 37
      4.4.2 Key software component designs.................................................................................... 40
   4.5 Pattern language....................................................................................................................... 42
5. Evaluation....................................................................................................................................... 45
   5.1 Evaluation method.................................................................................................................... 45
   5.2 Evaluation results..................................................................................................................... 46
      5.2.1 Supporting the artifact's potential..................................................................................... 46
      5.2.2 New requirements for the artifact.................................................................................... 47
   5.3 Requirement changes based on the evaluation......................................................................... 49
6. Findings.................................................................................................................. 51
7. Discussion.............................................................................................................. 53
8. Conclusions........................................................................................................... 54
References............................................................................................................... 57
Appendix A. Repository links & Demo................................................................. 62
Appendix B. Initial interface design................................................................. 63
Appendix C. Evaluation experiment tasks and questions............................. 65
1. Introduction

Originally patterns as reusable components and part of pattern language were introduced in the context of architecture by Christopher Alexander according to Pan and Stolterman, (2013). Reusing best practices has historically attracted more attention in software engineering than in HCI field according to authors. Pan and Stolterman (2013) present that patterns have been used in many other purposes in the field of information technology such as ethnographic research and ubiquitous computing. Their study finds out that eleven out of fourteen pattern language experts they interviewed view that patterns in HCI and design field have positive future. Pan and Stolterman identified that pattern languages to have numerous benefits based on the views of the experts. It was noted that patterns are good at things such as capturing and representing knowledge, documenting conventions and improving communication and creating form of communication for people with different roles and level of knowledge. Starting the research the hypotheses was that in order to usability patterns to be efficient to use they need to be easy to find, easy to connect with each other and be fast to deploy. Seffah and Taleb (2012) argued for the importance of maintaining and defining pattern interrelationships thus arguing for the importance of obvious connections between the patterns. They also stressed the importance of road between discovery and delivery of the patterns. Authors focus wasn't on the speed or usability however but making sure that patterns are presented in a manner that convinces the user. Second hypothesis was that one of the challenges is how to integrate patterns together with various amount of combinations of patterns. This issue was again discussed by Seffah and Taleb by description of the issue that universally accepted way to structure and categorize patterns doesn't exist yet. Lastly hypothesis was that patterns should be designed way that they aren't just visually designed to meet the desire of the designer or the developer but are easily customizable. Lindsey (2012) argued that issue with if framework that developer or designer works with isn't easily customizable or the documentation of usage is poor the end result of designs can end up with little differentiation with each other.

This study found out that even though there is numerous reasons why patterns create excitement in researchers there are also number of challenges in creating pattern languages. These challenges were then utilized in the design of the delivery platform artifact as challenges to solve. Second section of the literature review of the study focuses on the web platform and languages utilized by the platform creates unique challenges for implementing reusable solutions. Again the knowledge presented guided the design of the pattern language artifact and delivery system artifact.

Study reviews challenges that array of different devices from mobile phones, tablets, television sets and wearable devices employ to pattern design process. Nebeling and Norrie (2013) list that these devices present the content in different screen areas, different input types and screen resolutions which creates a need for adaptive design that can meet the needs of each device brings. They also mention gesture and pen based input as possible inputs and devices that act as interactive surfaces (Nebeling & Norrie, 2013). Responsive design (Marcotte, 2011) is at the time of writing a popular design method to tackle this issue by utilizing CSS3 media queries and other adaptive techniques discussed more in detail in the study. Queries are used to apply CSS rules to elements depending on the resolution of the browser window. (Nebeling & Norrie, 2013)
Current popular web front-end frameworks such as Twitter's Bootstrap and ZURB Inc's Foundation focus on delivering smaller individual components that web developers can reuse. It can be observed that at the time of the writing these frameworks provide little information about the context and usability notions of the components they provide and offer little variation by default. Lindsley (2012) examines Twitter's Bootstrap front-end frameworks shortcomings as an example as it was and still is at the time of the writing of this study the most starred repository in Github (GitHub, 2015). Bootstrap's success does however point out that there is a need for reusable components that play well together for the Web.

Larger emphasis in the latter part of the literature review is put on the evaluation of the patterns potential as tool for prototyping, solving the issue of accessibility and patterns role as persuasive actors.

Research questions for the literature review was how should platform for delivering pattern language for websites be build and what important requirements in pattern language. Method for the background research was narrative literature review. Focus was on summarizing the most relevant and highest quality knowledge available to answer the research questions.

Research question for the design science research is how should the pattern delivery platform and pattern language be build by accounting the previous shortcomings of pattern languages found in the literature. Shortcomings include in this context the found requirements and potential improvements found to pattern languages in the literature. Secondary research question is defined as how the developed system and the pattern language would fit to developer's workflow. After figuring out which kind of pattern language is needed based on the literature an evaluation are the practical needs to be conducted.

Literature review of this study features two parts. First part focuses on the merits and challenges of patterns and pattern languages in general in the context of HCI. The second part looks at the unique aspects of web front-end development and the Web itself that are evaluated to have effect on the development of pattern languages. Section also provides look on the recent developments of patterns outside the academic research.

Literature review is followed by description of research methodologies employed in the design science research. Next chapter describes and documents the building of the artifact of web-based delivery system for pattern language's. Delivery system is the intended main contribution of the thesis and is imagined to be used in the future as basis by anyone who desires to develop and distribute any pattern language. After the delivery system the second artifact, developed example pattern language is more briefly discussed. After the description the artifacts are evaluated with experiment with expert evaluators. Findings in the literature and their relation to evaluation results are presented. Finally the study ends with discussion and conclusion.
2. Background

2.1 User interface design patterns

Following section includes discovered highest quality knowledge about patterns and pattern languages in general. After defining the concepts and their background the section focuses on describing the benefits and challenges of patterns and their part in the design process.

2.1.1 Definition of user interface design patterns

Seffah and Taleb (2012) conclude that different definitions of patterns exist and they can even be contradictory to each other. Pattern can be defined as named, reusable and generally repeatable solution to commonly occurring issue. It can be formed by form, template, model or set of rules which can be utilized to generate things or parts of a thing. Specifically related to context of user interface designs pattern could be defined to be a solution to usability problem which occurs in varying contexts. Features of patterns include specific attributes and ability to be combined together with compatible pattern languages. Notable feature of patterns according to Seffah and Taleb (2012) is that patterns aim to provide concrete solutions to the problems while being abstract enough to be applicable in various situations. (Seffah & Taleb, 2012)

Dearden and Finlay (2006) define that in software engineering and HCI it is agreed that pattern is a structured description of an invariant solution to a recurrent problem in context. They note that this approach supports the Christopher Alexander's original problem oriented approach to patterns and that invariant solution means that solution offers set of characteristics that need to adapted to specific situation in which the solution is applied. (Dearden & Finlay, 2006) Authors note that there have been other ways of defining patterns by such as dividing patterns further to activity patterns and design patterns. Activity patterns describe pattern as they are without giving an opinion whether it should be used again in emulated manner or if it is worth preserving. Design patterns by this definition on the other hand focus on describing the problems relation to solution that has been proven in practice. (Bayle et al., 1998)

2.1.2 History of usage of patterns

According to Seffah and Taleb (2012) concept of design patterns was introduced in the late 1970s by architect Christopher Alexander. Alexander's books A Pattern Language and A Timeless Way of Building include knowledge and discussion of usage and collection of design knowledge in format of patterns. He presents collection of pattern examples that make us common language to guide architects and engineers to design buildings, towns, and other urban entities as defined by Seffah and Taleb. Alexanders patterns introduced solutions to recurring issues faced by designers and offered solutions within specific context. Seffah and Taleb also argue that key feature of Alexanders patterns were that they left room for designers creativity and allowed implementation vary based on design details. Seffah and Taleb note that Alexander's work on the concept of patterns have later influenced the utilization of patterns in object-oriented programming and inspired the thinking of capturing and reusing the design knowledge efficiently. (Seffah & Taleb, 2012)
Borchers (2000) argues that human-computer interaction can adapt the original patterns idea from the architecture unlike software engineering. He notes that there is a direct connection between design work and user experience similarly as Alexander noted that there is connection between the artifact that architect designs and it's inhabitants. In contrast Borchers notes that artifact that is created by software engineer designs is not something that users should have to directly interact with. Borchers also notes that another thing that is similar between originally introduced concept of patterns and HCI patterns is that they aim to be readable and understandable to both professionals and non-professionals. He again compares this convention to software engineering and points out that most successful book at the time of writing in software patterns community isn't readable to people that don't possess software engineering knowledge. Last difference Borchers points out is the lack of common values that bind the goal of the patterns together which is what what Alexander called “the Quality Without A Name”. Borchers hints that important to compare original architecture based patterns and software engineering patterns since pattern concept has been well know in SE research and practice the HCI publications cite Alexander's work as main influence. (Borchers, 2000)

Borchers (2001) suggests that idea of patterns in HCI research was introduced by Norman and Draper in classic Psychology of Everyday Things (1988) where Norman states that he was influenced by Alexander's work (Norman, 1988). Pan and Stolterman (2013) argue that big step in pattern language's influence to HCI was Erickson's suggestion of utilization of pattern language in interdisciplinary work. They note that even bigger influence was the Erickson's introduction of viewing pattern language as lingua franca between people with different expertise which have been later repeated often. (Pan & Stolterman, 2013)

### 2.1.3 Tools similar to patterns

Seffah and Taleb (2012) argue that guidelines and claims are two other methods of rounding up and distributing design knowledge. Authors note that guidelines have had the tendency to act as platform specific style guides that told the designers how windows should be styled and specific user tasks should look like. They summarize that guidelines concentrate mostly on the physical attributes of the user interface. (Seffah & Taleb, 2012) Hertzum (2010) suggests that the most extensive collection of general HCI guidelines consists of 944 guidelines compiled together by Smith and Mosier in 1986. Hertzum notes that issue that Mosier and Smith found out with their guidelines that user had hard time locating guidelines that were relevant to the system in question. (Hertzum, 2010)

Claims are design justifications that are based on psychological grounding that express the advantages and disadvantages of certain design as usability issue according to Sutcliffe (2000). Claims include number of situations of use and artifact that illustrates how the claim should be applied. Advantage of claims is that they make designers to see the tradeoffs of different solutions instead of focusing on single guideline. Disadvantage according to Sutcliffe is that they are tied to the situation provided by the artifact and the usage scenario. (Sutcliffe, 2000) Seffah and Taleb (2012) add that concept of reuse was later introduced to claims making them less dependable on application context.

Style guides are also a format used to communicate design experience to others according to Borchers (2000). He argues that style guides are efficient medium to use to ensure consistency with single user interface environment or toolkit. Disadvantage of style guides according to author is that they don't work well with different toolkit. Borchers notes that standards like ISO Multimedia User Interface Design Software Ergonomic Requirements are useful in creating rigid rules either in developing or
mature field. He suggests that when standards lack concrete examples and informal writing they aren't helpful resource to learn design. (Borchers, 2000)

2.1.4 Benefits of user interface design patterns

According to Seffah and Taleb (2012) benefits of utilizing pattern in user interface design include reusability of known best practices, being context and problem oriented and easy to use even for inexperienced designers. Patterns act also as basis of documentation of good design knowledge for designers. Patterns can also defined to be independent from specific toolkits. Seffah and Taleb argue also that patterns have the benefit of being a good communication tool due being clear and readable enough for not only to designers, but to developers and other stakeholders also. This view is shared also by Borchers (2000) who makes a claim that using patterns in software architecture, interaction design and the application domain of project at hand can improve communication between interdisciplinary teams. Patterns acting as communication tool and 'lingua franca' has been discussed also by Bayle et al. (1998) and noting that the fact that they are drawn from the situation for which design is being done makes them good at this. Seffah and Taleb argue that patterns can represent knowledge from different viewpoints such as social and organizational aspects and include views from different levels of design such as information from conceptual or detailed level view. Patterns form on based on practical knowledge instead of being created artificially according to Seffah & Taleb. (Seffah & Taleb, 2012)

Benefit that separates patterns from other methods according to Seffah and Taleb is that they are generic enough to allow different implementations to create variations in look and feel of the final design. Different implementations are easily doable within the limits of essence of the principles of good design introduced by the pattern. Two cardinal properties of pattern include firstly that patterns include user-centered values as their source of rationale. Secondly they argue that the concept of patterns and their associated pattern languages are generative, and can thus act as support for the development of complete design. (Seffah & Taleb, 2012)

Pan and Stolterman (2013) interviewed 14 HCI pattern language experts that according to them presented majority of the existing pattern language experts in human-computer interaction research. When asked from the experts about the advantages of using pattern language the resulting list is much inline with the assertions of Seffah and Taleb. Pan & Stolterman identified based on the interviews education, representing knowledge, documenting conventions and innovations and sharing which are concepts already argued in this study. They also identified the advantage of using pattern language as a tool for identifying interactions in HCI when compared with other methods of design. Authors also suggested based on the interviews that pattern language provides advantage in formation and advancement of design thinking. Pan and Stolterman explain that patterns can effect the thinking of the pattern's user in certain direction and to be careful and diligent with definitions and details. Lastly authors point out that pattern languages can also be seen as form of representation that promotes creativity.

2.1.5 HCI pattern language definition

Seffah & Taleb (2012) define pattern languages a structured way of defining good design practices which includes a collection of interrelated patterns that attempt to share the featured knowledge. They note that for designers the pattern languages describe the key features of effective solutions for reaching their design goals and communicative tool. Pattern languages also allow designers look for solutions for common HCI issues in logical way. Seffah & Taleb define three essential elements that pattern languages have. Firstly language has to contain a standard pattern definition. Secondly language
has to group the patterns logically. Finally the language should describe the interrelationships between the pattern. Seffah and Taleb argue that good characteristics of a pattern language also feature ability to walk the designer from pattern to pattern systematically. They suggest that pattern languages should be able to this way support the entire design process by being intimately related to it and guide the designer towards complete designs. (Seffah & Taleb, 2012) Borchers (2001) adds to definition the pattern's ability to look for the context that it can be applied from the descriptions included in the higher level patterns. Pan and Stolterman (2013) note that pattern languages have been used for many purposes as in interface design, ethnographic research and web usability. They also list the usage in many sub-disciplines such as ubiquitous computing, Human Robot Interaction and already noted software engineering.

2.1.6 Forming design process with pattern languages

Pattern are needed according to Seffah and Taleb (2012) because there isn't any viable concept for reusing existing designs and solutions to issues. This is due large variability in methods of design and lack of standards that would define a common set of rules for the designers. Their view is that in order to systematic user interface design and development to happen efficiently there is a need for a disciplined form of reuse. Systematic, traceable and practical tools that allow utilization of creativity are needed in the design process due multi-dimensional complexity introduced by design problems. (Seffah & Taleb, 2012) Borchers (2000) also suggests that patterns in various domains can be mapped to most phases of usability engineering life cycle outlined by Nielsen (1994) and can thus as communication tool throughout the process.

Seffah and Taleb (2012) argue in their paper that one of the big challenges in user interface design process is that the process involves often different participants and stakeholders. Challenge is to find common language for communication and sharing the knowledge between the participants. They see patterns as tool which overcomes these communication issues by providing a channel of communication that works across different cultural and professional backgrounds and satisfy the goals of different participants. They also argue that patterns are generally presented in readable format with high level of consistency and provide background information about the reasoning.

Seffah and Taleb defend in the paper that that commonly agreed UI design process that would employ pattern languages as main method of guiding the process doesn't exist. They suggest that pattern-driven design process should utilize the current issue of the designer to guide him towards relevant patterns. Furthermore the process should demonstrate how the patterns can be can be utilized and illustrate the possible combinations with related patterns.

Seffah and Taleb present three relevant design approaches from literature that are driven by patterns. First of the three is Pattern-supported approach (PSA) framework by Granlund, Lafrenière, & Carr (2001). Seffah and Taleb summarize that the key takeaway idea of the framework is that different parts of the design process can guide which pattern should be utilized, documented or identified. Second approach that the authors summarize was introduced by Duyne, Landay, & Hong (2002) which sorts the patterns into 12 groups which indicate the order which to apply the patterns in the process of Web design. Seffah and Taleb also point out that it is notable that pattern language introduces links to related pattern within the language. They describe also that process starts from highest level group Site Genres which presents the designer the option to choose which type of site he is creating and allowing him to move to lower groups utilizing the links to relevant patterns that are provided. Third design approach that Seffah & Taleb find notable is pattern-oriented design (POD) which has been
proposed by Javahery & Seffah (2002) which aims to make it obvious to developers how to combine patterns with each other. Authors describe that first of two main features of POD is that it provides design steps where set of patterns are applicable and this way creates systematic process which to follow. Secondly authors describe that in order to help designers to combine the patterns the pattern relationships are explicitly described. Seffah and Taleb present five different relationships that they think can be easily applied to other pattern languages which are similar, competitor, super-ordinate, sub-ordinate and neighboring.

Seffah and Taleb also present that recently there have been suggestions from authors to use patterns to supplement preexisting design methods such as model-driven and component driven design. They present the concept that after choosing a pattern software component can be chosen to implement the pattern instance.

2.1.7 Utilizing user experience knowledge with patterns

Seffah and Taleb (2012) argue that pattern languages have neglected to provide information about the users and their user experience. They describe this is a major issue that prevents patterns from reaching their original premises of offering solutions for usability-specific problems in human-computer interaction design. Seffah & Taleb explain that pattern languages tend prioritize the information about the users low and argue that to keep the focus on usability the end-user perspective should be key part of elements that form the pattern. Seffah & Taleb introduce the discussion of Welie & Veer (2003) and Welie & Trætteberg (2000) about proposing that the patterns should be presented from end-user's perspective with the focus on usability aspect of design. They suggest that featuring information about user problems and deriving the information from understanding the user experiences.

Welie and Trætteberg (2000) suggest that patterns should describe successful solutions that aim to benefit the user of the system. They suggest that pattern creators should note that while each pattern that focuses on benefitting the user is also usable for the designers but same doesn't necessarily apply other way around. Authors point out that some patterns might set out to solve problems of designers or marketers instead of user citing an ad banner as example of design that doesn't increase usability of user but is considered as accepted design. Authors point out that acknowledging the view point is important in order to avoid situation where one is looking at the problem from the point of the solution without knowing the actual problem for example when design is out of context.

Welie and Trætteberg argue also that each pattern should utilize ergonomic user interface principles that leads to improvement of the usage indicators. In their pattern language they utilize user interface principles suggested by Donald Norman in the book The Design of Everyday Things. Welie and Trætteberg utilize these principles as a way to categorize users problems. They list following principles; visibility, affordance, natural mapping, constraints, conceptual models and feedback as principles that assume that users always exhibit rational behavior. Additionally they add principles of safety and flexibility as principles that take into account that users make mistakes or change their minds. Authors also continue to list following usage indicators that can be impacted by utilizing patterns; learnability, memorability, speed of performance, error rate, satisfaction and task completion based on the study by Welie, Veer, & Eliëns (1999).

Seffah and Taleb define user experience as term that consists of collection of information about user behaviors, expectations and perceptions. These attributes are according to their definition further influenced by application and user characteristics.
They define that user characteristics include user's knowledge, experience, personality and demographics. Furthermore they define application characteristics to include domain, content, language, visual design and interaction type. They suggest also that context of use can have effect on user experience through technical, social and organizational characteristics. (Seffah & Taleb, 2012) Welie and Veer (2003) also suggest that user experience isn't just about the tasks and goals of the user but also about how user reaches his goals using the concept of website, how they perceive the website and whether it gives the user the appropriate satisfaction. They sum that experiences are high level goals for which the users come to a website.

Seffah and Taleb suggest a method of persona to pattern framework to create possibility to guide designers to choose patterns based on the user attributes at hand. Personas is defined by Javahery, Sinnig, Seffah, Forbrig, & Radhakrishnan (2007) in their study as user description in the form of variables with discrete domains. They explain persona is formed by characterizing each representative user by set of variable assignments. Seffah and Taleb argue that data should be collected to create set of personas that illustrate the user experiences of existing or imagined system. Seffah and Taleb describe that in their study they utilized heuristic evaluation and psychometric assessment to add more empirical data to their persona descriptions. They extended the personas to include information about interaction behavior, user tasks and goals. Representing the personas in discrete way to correlate information to patterns makes them suggest that user types that have different needs relative to design can be identified. This leads to ability to offer improved usability to those users according to authors. They also note that determining usability problems that apply in certain applications should be made possible to support the design process. (Seffah & Taleb, 2012)

2.1.8 Pattern semantics and representation

Seffah & Taleb (2012) argue that the main goal for patterns is to be forwarded to users who then implement them as solutions. They explain that for this reason it is important that pattern will be represented in a way that software developers are able to learn, master and easily and effectively apply patterns in their context. Another argument from them is that this sort of implementation is important for the main role of the patterns which is to promote effective reuse. Seffah and Taleb explain that pattern should feature information that supports the suggested solution in order to convince the user that the solution is worth implementing. They mention that this could be helped further by adding expert analysis and comments, example cases where the solution has been applied and even some code examples. Furthermore they argue it is important to include the guidance how the pattern should be applied in different contexts and situation's but room should be left for designer to adapt the pattern in new design solution.

Seffah and Taleb (2012) explain that potential benefits that patterns provide for reusability doesn't necessarily remove the designers efforts for creating the design completely. They argue that requirement to modify existing designs for reuse is that designer is experienced and be ready work considerably to reach this. Lindsley (2012) on the other hand discusses the issue that when novice developers and designers utilize frameworks like Twitter's Bootstrap they can end up with designs with little differentiation to each other. He suggests more guidance for the novices is needed to achieve applications that fit to user needs and specific contexts of use. Lindsley argues that source code should be thought about as interface as well, one that should not leave the customization ability to experts and framework developers but enable and educate the customization benefits and options to all users. (Lindsley, 2012).

Seffah and Taleb (2012) add that in order to design ideas to become reusable they need to be packaged and abstracted into suitable formats. This is helpful also to allow the
software designers keep some structure in their process, control their design activities and work within available resources.

2.1.9 Issues with development of pattern languages

Seffah and Taleb (2012) argue that one of the current issues is that in HCI community hasn't agreed upon structured and formalized use of patterns. They argue also that there isn't universally accepted way to categorize patterns in pattern languages. Seffah & Taleb explain that most serious defect in pattern languages is that pattern interrelationships are presented incompletely and lack the context oriented perspective. They add that interrelationships affect the patterns context of use by giving the designer set of circumstances in which the pattern is applicable. Seffah and Taleb furthermore list that among the current issues that no standard of documentation exist within each pattern language. They assert that there also isn't commonly recognized way to structure pattern languages which indirectly reduces the usability of the languages for the designer. (Seffah & Taleb, 2012)

Seffah and Taleb argue that issue with the documentation is also that at the time languages use narrative text formats with varying attributes which can cause misunderstanding and confusion. This also causes according to them that pattern languages are difficult to use, especially when user attempts to combine patterns from different pattern languages. They also compare pattern languages to other languages in computer science that they share similar syntax and semantic that is missing from the pattern languages at the time of writing. (Seffah & Taleb, 2012)

Pan and Stolterman (2013) list challenges in developing pattern languages found out by interviewing pattern language experts. First viewpoint they mention is that developing pattern language requires lot of work with quote of expert saying that basic development can easily consume more than year of one's life. Secondly authors list suggestion that patterns can be too formal and hard to organize. Following formal structure imposed by patterns can be seen as restriction by designer according to expert suggestion. Expert quote in the study suggests that it can be difficult to find pattern suitable for the situation if the pattern language lacks good structure and organization. Third challenge that Pan and Stolterman highlight is the issue how to get people to adopt pattern languages in real world setting which they mention being even more challenging than developing them. Last challenge presented in the study is the suggestion that patterns can be seen as in-between constructs between abstractness and specification. Expert interviewed sees patterns as generalization of multiple examples and situated in between abstractness provided by theory and specific instance of example. (Pan & Stolterman, 2013)

Pan and Stolterman (2013) also interviewed pattern language experts in order to figure out the reasons why pattern languages haven't been successful despite of the excitement of their promise surrounding them. Authors point out the formalization and extensive efforts required to build pattern language which were mentioned as challenges also as reasons for this. Expert is quoted suggesting that designers resist more descriptive formalizations than generative or process-oriented ones like design methods. Extensive efforts in collective work, sufficient experiences and knowledge is required in order to develop a good pattern language that provides worthy insights according to authors. Pan and Stolterman pick out lack of validation and evaluation of pattern languages as the third reason based on the interviews. They point out that there is lack of empirical research which aims to evaluate the effectiveness of pattern languages in practical design settings. Lastly authors pointed out that several experts mentioned the issue of dynamic nature of technology and design. They quote expert arguing that lack of
staticness means that value of the patterns will always be very short and pointing out that this is especially true in the field of interaction design.

Pan and Stolterman stated in their study that eleven out of fourteen experts think that there is positive future for the pattern languages in HCI and design field. Authors suggest based on the expert views that focusing on creating pattern languages instead of incompatible pattern and more empirical research for the practical effectiveness of pattern languages are keys to pushing positive future. Even though this positive outlook by the experts Pan and Stolterman conclude their study with the prediction that it is highly unlikely that pattern languages will soon become successful and widespread practical method for interaction design practice. Authors based this prediction to their observation that pattern languages that have been authored haven't been deployed on a larger scale.

2.2 User interface pattern challenges in the Web environment

Following section focuses on presenting highest quality literature about challenges that the web environment creates as platform for user interface patterns. Section also focuses on describing recent practical approaches to pattern languages in web environment. Potential uses for patterns as delivery platform for accessibility, cultural and persuasive knowledge is also presented.

2.2.1 Benefits and issues of responsive design

As outlined in the introduction section of this study the one of biggest new challenges in modern web development is the changing landscape of different devices that are capable of rendering web pages. Mobile landscape has changed away from feature phones with Jakob Nielsen pointing out already in 2011 that it might not be wise to support them due poor usability and low usage in browsing the web (Nielsen, 2011). Responsive design term is often credited to Ethan Marcotte's web article published in 2010 where he first put the concept under the term (Marcotte, 2010). Marcotte outlines this technique further in his non-academic book in which he describes it as technique to design web pages that adapt to different contexts by utilizing media queries, fluid grids and flexible images (Marcotte, 2011).

Bridges & Rempel write about their experience of replacing mobile site with responsive design and conclude that responsive design frees developers from maintaining separate versions of both mobile and desktop pages. They mention fact that only one version of the site is needed it is simpler to create iteration that meets accessibility guidelines (Bridges & Rempel, 2013).

In context of developing native mobile application utilizing responsive web design methods Pandey (2013) notes also the benefit of single and uniform code base that is lucrative in a way that it helps to reduce development costs and maintenance costs that follow. He also notes that responsive design choosers need to aware of caveats of performance, code complexity and lack of able to deliver targeted and device specific experience for the users (Pandey, 2013).

There are also plenty of practical development issues introduced when attempting to support multiple devices. Responsive images one of the main issues that still doesn't have widely accepted solution according to Matsudaira (2013). Matsudaira defines responsive images solution to problem of delivering right image to right device. He explains that Apple's solution to high resolution screens at the time of the writing of the study was Javascript based solution to replace regular images with higher resolution ones when device with high resolution density was detected. Matsudaira outlines also
number of factors that can impact the user experience of mobile devices that are unique to them and need to be accounted. These factors are limited screen size, smaller number of simultaneous connections, slower network, slower processing power, smaller caches and incompatible browsers. (Matsudaira, 2013)

Pandey (2013) thinks that responsive design might not work as best possible option for providing best user flows and experience in apps that require complex workflows and when user benefits from interactions that are adapted to suit the device at hand. He argues that in these kinds of scenarios simply remixing and reordering the content with same interaction strategy might not be sufficient.

2.2.2 Recent approaches in pattern languages for the Web

Many of the practical pattern languages for the modern web development seem to come from individuals working in the industry or released by companies and are build without academic evaluation. Atomic Design is a pattern based methodology outside academia to building web design systems introduced by Brad Frost and Dave Olsen (2014). Atomic design categorizes components used in web design into five categories where elements from the lower level categories can be used to form elements into categories above them. Atomic designs taxonomy consists according the website description from following categories in order of bottom to top; atoms, molecules, organisms, templates and pages. They define atoms as smallest building blocks that can't be broken down and include in practice basic tags such as form labels, inputs and buttons. They also include abstract elements such as color palettes, fonts and animations into atoms. Molecules in Atomic Design concept are defined as group of elements that function together as a unit. As an example the authors cite that a form label, search input, and button atom can be combined to form a search form molecule. Organisms are defined as groups of molecules and possibly atoms joined together to form distinct section of an interface. Frost & Olsen cite as an example of organism being a masthead organism that might include logo, navigation and search form. Authors define templates as mostly compromising organisms combined together to form page-level objects. Focus is on defining content structure and giving less abstract context than molecules and organisms in order to communicate with other designers and clients according to Frost and Olsen. Lastly they define pages as being instances of templates that utilize real representative content instead of placeholders and being aimed as tools for evaluating the effectiveness of the design system. (Frost & Olsen, 2014)

Web Components is combination of standards currently in production at the time of the writing by W3C with the goal of creation of reusable components for the web. In the current form it consists of four pieces; Custom Elements, Shadow DOM, HTML Imports and HTML Templates. (Cooney & Glazkov, 2014) Adoption of Web Components could in the future lead to a new level of pattern usage adoption by web developers since as a standard it is likely to be supported by browser vendors and thus natively by browsers. Despite the premises there appears to be little academic discussion about the potential.

2.2.3 Issues with adoption of tools in web development

Lindsley (2013) brings forward the notion that because of the lack of common conventions in web development frameworks have become the source for standardization of common web programming practices. He adds that most frameworks also offer plug-ins or an ability to utilize code distributed by other developers that works with existing code of the framework. Lindsley also cites thoughts of Paul Scrivens of fear that using development tools becomes synonymous with programming knowledge.
Lindsley's arguments become relevant when considering the deployment of another framework to the current web development landscape.

According to Harper and Chen technology adoption occurs when developers see that there is a good reason to make a technical change. They also make interpretation based on their study that open technologies have better chance of because the browser vendors can natively build in the support to them. Harper and Chen refer to Javascript adoption by browser vendors. (Harper & Chen, 2012)

### 2.2.4 Challenges in web pattern interpretation

Islam (2013) defines objects that convey web content and system functionalities such as navigation links, buttons, icons, labels, thumbnails and symbols as interface signs in his study. He argues that there are number of factors that affect interface signs intuitiveness and if user interpret the signs meaning similarly as other users. Findings of the study present that determinant factors that affect are amplification features of interfaces signs, interface sign position, sign color and matching features of interface sign.

Amplification features of interfaces signs are as defined by the author as features that support and help users to interpret meaning of other signs. Islam lists as possible amplification signs being such as appended small image, appended thumbnail, appended icon with textual sign, appended short text, appended indicative text with iconic sign or appended abbreviated letter with textual sign.

Interface sign position was found out to affect user's interpretation through effect of familiarity of the position of the sign and neighboring signs. Familiarity of position helps because the user is used to seeing certain signs in specific locations of the web interface. Neighboring signs help by giving context that helps the interpretation and might even be the key element that enables the user to understand the meaning of the sign according to Islam.

Sign color can help users to interpret interactivity or in practice if the sign is currently active or inactive and if it is currently visited or already visited. Islam argues that color usage has positive impact to sign interpretations and cites conducted study as an example that if brand color is missing the user is less likely to recognize the brand.

Matching features of interface sign on the other hand deals with the notion that user's interpretant need to match with the interface sign's object in order correctly interpret the meaning. Islam suggests attributes such as conventions, underline reality, clearness of designer's motivation, correspondence with the real world object and functional similarities that can act as matching feature. Islam also highlights the importance of the user's presupposed knowledge which he refers as pre-familiarity of interface signs and their meaning in the context of use. (Islam, 2013)

Islam's research points to importance of having multiple signs that clarify the meaning behind intractable and non-intractable content on the web. Specially it can also be interpreted as pointing out the importance of icons usage and using them correctly on the web. Huang, Bias,and Schnyer argue based on their study conducted by utilizing fMRI data of how people read and interpret symbols and icon that icons aren't as efficient as words in conveying semantics. Their study finds that icons are processed in the brain similarly to pictures and require more brain resources in order to interpret the meaning. Authors also argue that brain resources are also needed to learn the meanings of the icons and that additional context is often needed for the interpretation.
Shortcoming that can be seen in the study by Huang, Bias, & Schnyer (2014) is that they studied the interpretation of icons without the context which as shown by Islam (2013) can have major impact. Icons presented to subjects in the study also lacked color which can also have impacting factor to interpretation according to Islam's study. Huang, Bias and Schnyer argue that their study is further proof that stand-alone icons have no superiority over texts in terms of usability. They argue that since Wiedenbeck's (1999) common belief has been that icons are more efficient and effective than text. Wiedenbeck did however state as main conclusion of her study that icons lacking labels perform very poorly for the first time users. She also stated that labels play important role in the learning of icons meaning of the users who had little knowledge of their representational conventions. Recommendation from her was that text labels alone or icons paired with labels should be presented to learners of applications. (Wiedenbeck, 1999)

Cultural background has also been shown to impact the interpretation of the web as argued by Reinecke and Bernstein (2013). Authors initial argument based on the literature is that while most localization on the websites is limited to language, date and time formats, there is possibility of significant gains by in utilizing more comprehensive modifications that cater users cultural background. Their proposal in the study to address this issue is culturally adaptive user interface which adapts to users preferences once it recognizes his cultural background. Reinecke & Bernstein identified number of cultural influences which to base these UI adaptions; language, reading & writing direction, religion, political orientation and social structure, education level and form of instruction received in educational institutions. Authors argue that UI should take these influences into account and adapt in number of ways such as changing the alignment of objects based on the reading and writing direction preference, varying information density based on social structure and different colorfulness and contrast based on religion.

2.2.5 Patterns as tool for solving accessibility issues

Harper and Chen (2012) define web accessibility as practice that aims to help people with disabilities to perceive, understand, navigate, interact with, and contribute to the Web. They assert in their study that accessibility guidelines and recommendations are adopted at a significantly lower rate than other mainstream Web recommendations and technologies. Harper & Chen note that while screen reader can attempt to offer ways to navigate and interpret the content of the Web the richness is lost due visual nature of the Web. They point out that currently the most promising set of guidelines Web Content Accessibility Guidelines version 2.0 is unlikely gather much larger adoption than the previous ones. Authors argue that while guideline includes useful techniques for testing the conformance of the guidelines the issue is that most of the procedures require manual evaluation and repair from the developer itself.

Harper and Chen interpret that Accessibility Guideline adoption has stagnated at less then 10% over ten years which means as a result that most websites have accessibility problems that their usage for people with disabilities ranges from difficult to use to impossible. In their study Harper and Chen conclude that in order to Web accessibility standard adoption to improve significantly the guidelines should be baked into technical language specifications. Authors remind that the core issue issue is that with front-end web technologies there is no control of the predicted outcome in a way that exist in homogenous technologies such C or SQL where there exist developer led conformance. (Harper & Chen, 2012)

Fogli, Parasiliti, Provenza and Bernareggi (2013) also criticize both WCAG 2.0 and WAI-ARIA reference specifications by W3C for accessible rich internet applications.
They point out that both specifications contain so much information that it discourages web designers from dealing with issues related to accessibility. Authors argue that guidelines are utilized more in criticizing the finished design rather than how to solve design problems during the design process. Fogli, Parasiliti Provenza, & Bernareggi propose pattern language for accessibility that utilizes the knowledge from WCAG 2.0 and WAI-ARIA specifications and assessed by users. They note that existing design patterns that deal with accessibility are focused on the static aspects of Web sites and suggest constraints to interactivity as an example recommending to not use moving objects. Authors argue that such policy isn't feasible in current web applications that utilize scripting languages to deliver rich interactions for the users. Another issue according to them is again the missing interrelationships between the patterns. Fogli, Parasiliti, Provenza and Bernareggi also aim to improve the existing pattern solutions by structuring bridging patterns that include information how to implement the solutions proposed by WCAG 2.0 and WAI-ARIA.

Authors list four main accessibility challenges created by rich internet applications. First challenge is dynamic update of web content in different areas of web page. Second challenge is created by interactive elements on a web page that do not receive focus when navigation using the keyboard is utilized. Third challenge is lack of semantic structure in page areas defined by XHTML. Fogli et al. point out that if role of the page are or goal of the functional element isn't defined the assistive technologies cannot provide semantic descriptions or goal of a functional element to user. Fourth challenge comes from custom controls such as drop-down and fly-out menus, sliders, tree menus and tooltips which do not have any standard mechanism that would make them accessible. (Fogli, Parasiliti Provenza, & Bernareggi, 2013)

2.2.6 Patterns as a tool for prototyping for the web

Prototype can be build in software development context generally as user-interface that has minimal functionality, but provides provides feedback during development process (Soutome, Ling, Niibori, & Kamada, 2013). Borchers (2001) suggested that patterns can play important role in prototyping by using them to form architectural standards or project ideas. He also suggests that patterns could be used as basis for iterating the prototypes themselves. Following literature review looks at what features are expected from modern prototyping workflow in order to see how patterns could fit into this.

Pandey (2013) argues that tool or framework used for the prototyping should be able deliver mockups of the design ideas most importantly fast and should be easily shareable with the client. He adds also thought that if client needs learn to use or acquire tool or know-how to evaluate the prototype it reduces the prototypes value. Pandey shares the experience of utilizing Twitter's Bootstrap and Zurb Foundation web frameworks for prototyping mobile application and notes that they felt they weren't the best tool for the job. This was due because the web components needed much altering to look and feel in order to be suitable in mobile app context. He notes even with mentioned limitation, he thinks that frameworks offer invaluable insights and possibilities for responsive design for the mobile web (Pandey, 2013).

Soutome et al. (2013) argue that interaction to provide detailed info and transitioning between pages is important for purpose of evaluation of website prototype and its look and feel. This is due increase in cost of changes once the website has been built. They present an opinion that most cost efficient way to gain feed back during the development process is to present mockups and site maps. Major downside according to them in tools for creating website mockups that existed at the time of writing is that they aren't optimized to obtain feedback from the client and user is required to use different tool for the task. Soutome et al. propose development of web based prototyping tool that
allows clients to participate in evaluation of prototype in early stage by enabling direct commentary to mock-up and even possibility to make adjustments to the mockup.

Teixeira, Saavedra, Ferreira, Simões and Santos (2014) point out that it can be beneficial if the prototype appears to the user as being disposable in explanatory stages of development in order to demonstrate the early stage of development and ability to discard the prototype. They point out that purpose of this kind of low-fidelity prototype is to increase quality of requirements document and is used figure out the most complex requirements or used to obtain early feedback in uncertain environment. Evolutionary prototyping technique on the other hand focuses on the portion of the actual system where the requirements are well understood and requirements are incrementally added along with the development. Teixeira et. al evaluated web-based prototyping tool Lumzy that enabled also non-technical users to experience, test, collaborate and validate the simulated program over the Internet. Users were able to not only annotate the prototype but actively modified the interface according to their needs using the web-based tool. Role of the analyst was to collect the new requirements when notified and then form data model based on combination of requirements with information of elementary tasks. Study notes that prototyping tools that provide similar abilities are very appropriate to use when system in question is complex and dynamic by its definition. (Teixeira, Saavedra, Ferreira, Simões, & Santos, 2014)

2.2.7 Patterns as a tool for persuasion for the web

Hasle (2011) defines Persuasive Design as research approach that aims to study systems which attempt to change attitudes or behavior or both without resolving to use coercion or deception. He defines that Persuasive Design overlaps with many well-established ICT-disciplines like human-computer interaction, usability, interaction design and information architecture but takes its lead mostly from literature of social psychology (Hasle, 2011). Persuasive Design research can be seen as important source for theories that patterns can be build upon since as pointed out by Chu, Deng and Chuang (2014) along side with Hasle (2011) that every website has persuasive aspects in them. Oduor, Alahäivälä and Oinas-Kukkonen (2014) presented persuasive software patterns for social influence and argued for new research track of persuasive software patterns. Chu, Deng, & Chuang furthermore argue that Persuasive Design concepts can help designers to clearly define website's persuasion objectives and purpose. Authors point out that there are already multiple significant listings of principles of persuasion. Chu, Deng, & Chuang introduce firstly Cialdani's six principles of influence; reciprocity, commitment and consistency, social proof, scarcity, liking and authority. Fogg's (2002) persuasive framework includes over forty principles and has been divided to tools, media and social actors. Persuasive System Design framework by Oinas-Kukkonen and Harjumaa (2009) has been build on top of Fogg's persuasive framework and consists of 28 persuasive design principles. (Chu, Deng, & Chuang, 2014)

Chu, Deng, and Chuang (2014) identified 15 prominent and often seen persuasive triggers utilized in six most popular shopping websites in Taiwan. They used 33 persuasive tactics for Web investigation found from literature to match and identify the patterns in use. Authors argue that as the result of the review of the websites it was revealed that persuasive trigger can vary largely in its size, form, complexity and span. They point out that trigger can exist in a form of a single icon, a button, a label or a line of text or it can consist of combination of multiple interactive elements. Authors explain that persuasive trigger can also be a mechanism that spans across several web pages and that same design element can be used to implement different persuasive tactics at the same time. Based on interview of 12 users authors evaluated that usability, product information, premium, scarcity, social conformity, visual appeal,
tailoring/personalization, reassurance, and reliability were the persuasive principles that could have persuasive impact to users.

Chu, Deng, and Chuang note also that persuasive tactics and their instances might have varying effect based on the what the product type in e-commerce, product price levels, user characteristics and stages in the consumer decision cycle. Authors give examples of this in practice such that social proof trigger was evaluated as valuable tool for promoting products such as mobile phones and digital cameras since it can be interpreted as form of testimonial for the product. Chu, Deng, and Chuang suggest that similar social proof may result in opposite reaction in products that manifest individuality such as clothes in which case argument is that shoppers don't like to wear same kind of clothes as others.

Chu, Deng, and Chuang concluded their study results by discussing that wide product selection, fast and accurate product search, easily used user interface and both security and privacy protection seemed to be minimum requirements for establishing positive attitude for the web shop and assert them to be comfortable with the online shopping experience. They note that however superficial enticements cannot be expected to persuade users to make more purchases or revisit the site but instead aim should be to provide more value to the customer.

Kaptein, Markopoulos, de Ruyter, and Aarts (2010) argued that ambient persuasion could be utilized to target persuasion to specific context or specific individuals and present model for ambient persuasion. They propose that ambient persuasive technologies could be utilized to act as actor that can leverage context awareness, user awareness and situational awareness to improve efficiency of persuasion.

Gamification is another research topic with persuasive background that could be seen source for theories which to base interaction design patterns. Deterding, Dixon, Khaled, and Nacke (2011) define gamification as term for the use of elements from video games in non-gaming to improve user experience and user engagement. Key differentiation according to authors in gamified applications to actual games is that software application only includes elements from games and not an actual game in its definition.
3. Research Method

3.1 Design Science and Information System Research

Hevner, Salvatore, Jinsoo, and Sudha (2004) define design science's goals as aiming to create artifacts which can be used to solve specific organizational problems and evaluating those artifacts. Authors introduce the relevance of design science research by noting that importance of design in IS research has been recognized in the previous literature. They also argue that IS research has been previously linked to gain its value based on the applicability of the research.

Conceptual framework that Hevner et al. (2004) present is purposed for understanding, evaluating and executing information system research combining design-science and behavioural-science paradigms. Authors argue that this is an effective way to approach solving fundamental issues faced in the area of productive application of information technology. Depiction of the framework is presented in figure 1.

![Figure 1. Framework combining design-science and behavioral paradigms.](Hevner et al., 2004)

Key starting point for the framework for design science research that the Hevner et al. (2004) introduce in the study is the idea of complementary cycle between behavioural science and design science. They define the purpose behavioural science research as developing and justifying theories which explain or attempt to predict human or organizational phenomena among the analysis, design, management, implementation
and use of information systems. Behavioural science research aims according to authors explain phenomena which occur due use or intention of use of an IT artifact. Furthermore aim is also to explain phenomena related to IT artifact's perceived usefulness or impact on individuals and organizations which also have dependance on system, service and information quality. Authors define that on the other side of the cycle resides the design science which aims to build and evaluate artifacts instead of focusing on phenomena caused by them.

Research born from the design and evaluation process provides answers to the problems addressed by the artifact according Hevner et al. (2004) are often business related in their nature. Cycle formed by this is concluded by authors to be that firstly artifacts created by utilizing design science research are build upon the relevant behavioural science theories. Secondly behavioural sciences interact with phenomenas created by those technology artifacts by anticipating and engaging them. Authors define that the environment defines the problem space and affects the IS research context by expressing the business needs which the research can be based upon and thus brings relevance. IS research context on the other side can contribute to the environment by offering research knowledge which is applicable in appropriate environment. The knowledge consists from foundations and methodologies that have been born from prior research and in practice mean theories, constructs, models, frameworks, instruments and instantiations.

In addition to the conceptual framework, Hevner et al. (2004) suggested seven guidelines aimed towards researchers, reviewers, editors and readers of the design science research articles to allow them to better evaluate what are the requirements of effective design science research. Hevner et al. (2004) introduced their guidelines for design science research as being adaptive and process-oriented due the necessity brought by the constant scientific revolution that exists because of the technological advancements. Their fundamental principle for the guidelines is that understanding the design problem and knowledge related to it can be gained through building and application of an artifact. Authors described seven guidelines in following description:

1. **Design as an Artifact:** First guideline introduced by the authors states that the result of the design-science research must be artifact that is either construct, model, instantiation or a method and answers to important organizational problem. Authors highlight that it is important that artifact is described in a way that it can be implemented and applied in the designated domain.

2. **Problem Relevance:** Second guideline states that is that research must answer relevant and important problems faced by people in the context of business and in the area of the interaction of people, organizations and information technology.

3. **Design Evaluation:** Third guideline states that design artifacts utility, efficacy and quality has to be rigorously demonstrated by utilizing evaluation methods that are well executed. According to authors evaluation requires definition of appropriate metrics and likely collection and analysis of relevant data. They point out that typically methods for evaluation come from methodologies available from knowledge base.

4. **Research Contributions:** Fourth guideline according to the authors is the argument that design-science research must be able to clearly point out the
research contributions. Authors point out three types of research contributions that can be based on novelty, generality and significance of the artifact that is designed. Contribution areas that authors list are the design artifact itself, the design foundations and methodologies for the design.

5. **Research Rigor:** Fifth guideline argues that rigorous methods must be used to both building and evaluating the artifact. Often the rigor is comes from skillful utilization and selection of methodologies of the knowledge base according to authors.

6. **Design as a Search Process:** Sixth guideline states that design should be a search process to discovering an effective solution to a certain problem which includes finding and applying the knowledge from both application and solution domains.

7. **Communication of Research:** The seventh guideline states that communication of research and its results must be presented to not only to technology-oriented audiences but also to management-oriented audiences. Authors argue that managerial audience benefits from knowing what kind of knowledge is needed to realize the artifact in question.

### 3.2 Adopted Design Science Research Methodology process

#### 3.2.1 Methodology

Peffers, Tuunanen, Rothenberger, & Chatterjee (2007) presented methodology which aims to provide a framework for conducting design science research in rigorous manner. Their paper aims to provide principles, practices and procedures to carry out design science research in a process that accepted among researchers. Peffers et al. (2007) suggest also process model and mental model for design science research. Their methodology includes six activities which were born based on examining seven representative papers. From these papers Peffers et al. (2007) combined the common agreements into their research process. These six activities are problem identification and motivation, objectives of solution, design and development, demonstration, evaluation and communication.

![DSRM process](image)

**Figure 2. DSRM process (Peffers et al., 2006)**
Peffers et al. (2007) stated that the described process model that they present is structured in sequential order only in nominal manner. Authors point out that they don't expect researchers to follow the process sequentially and suggest that researchers could start the process from any of the four first activities of the process. In the case of problem-centred initiation of research the process the researcher would start from problem identification and motivation activity. When researcher gets started with objective-centred solution the starting process activity is objectives of solution as pointed by Peffers et al. (2007). Design- and development-centred approach on the other hand would begin with activity design and development. Finally the client/context initiated approach is predicted by the authors to begin from demonstration activity.

3.2.2 Implementation of DSRM methodology

Problem centred approach

Motivation for the study originally stemmed from author's identification of practical issue in web development of repetition. Author observed web developers implementing components and solutions over and over again instead of referencing to solutions that would give building blocks to build the already solved issue faster. Literature review introduced earlier in the thesis revealed potential of pattern languages as being possible solution to this issue and also brought up issues in current pattern languages that could be attempted to be solved.

Problem identification & motivation

Problem identified by this study is to find out if its desirable use pattern language based on web-based platform which implements structured taxonomy and interconnections in order to reduce repetition in user interface building in web development. Solution to this problem requires components of the web-based platform and the patterns themselves.

Objectives of solution

Objective of the produced artifacts, the delivery platform and the patterns is to demonstrate how interconnections and the taxonomy of the patterns which are freely available to developers can improve the web development workflow. Objective is to tackle also other criticisms of previous pattern languages such as lack of context definition and persuasive design suggestions for the patterns. Suggestion of the artifact will be mostly based on suggestions provided by literature discussed earlier in the study. Solution is expected to offer faster and more usability focused workflow for web front-end development process. Value of this is aimed be demonstrated to web development community by releasing the artifacts free open-source software and demonstrating the capabilities.

Design & development

Creation of the artifacts aims to follow best practices of software development by building on top of components that are already following them. Patterns developed will be based on real-life examples with references included on similar implementations. Created artifacts will also be documented in this thesis and more in detail on
development repository and delivery platforms website in order it to be accessible to everyone.

Demonstration

Potential of the developed artifacts are demonstrated by using group of selected users which have sufficient experience. Finally the artifacts are intended to be released publicly for web developers to use freely.

Evaluation

The delivery platform and the pattern are evaluated using experiment featuring qualitative survey with experts with experience in front-end web development.

Communication

Communication of this solutions importance is intended to be communicated through this thesis in form which is familiar to researchers and by releasing the developed artifact as open source software. Communication with web development community is also key to understanding the significance of released artifact in practice.
4. The Software Artifact

4.1 Motivation and contribution

In order to reduce the amount of duplicate work these usability patterns should be provided for the users in format of production ready and yet easily customizable code instead of static designs. Argument is that issue with duplicate work in web design is still present in process which design and wireframing is done in a way that doesn't produce any code and requires step of replicating the design plan in code. Emerging solutions enable designers and developers to draw the design as before and also have the semantic, readable and reusable code generated. Presenting the patterns in code enables developers to build sites faster as they have solid foundation which to work upon instead of starting everything from clean slate. Seffah and Taleb note that patterns should be presented in a way that makes them efficient to learn, master and apply effectively in their context of use (Seffah & Taleb, 2012).

4.2 Requirement Specification and objectives

Argument can be made that artifact of delivery platform should be build in a way that it enables designers and developers to quickly and effortlessly navigate different patterns and discover the links between different patterns. Ability to distribute to large scale of users, availability to users regarding the operating system and device and ability to upgrade the individual patterns and their descriptions efficiently are features that are desirable to reach efficiency. Pattern library should also be searchable through names of the patterns and tags that define the purpose and nature of the patterns in order to enable users to filter patterns by criteria. Platform should also be able easily support multiple levels of navigation due different taxonomies that can be utilized to categorize the patterns. When implemented as part of the web page patterns can be implemented and presented on the web page as they normally would instead of providing just image.

Due these requirements utilizing the web platform a choice worth evaluating for system which aims to present large number of pages with high amount of interconnections. Presenting delivery platform as web pages fulfils the mentioned requirements and also keeps the control of releases to platform owner. Kruschitz and Hitz conclude that most of the HCI design patterns are published through websites and major issue is that authors stop updating and maintaining the collections. They also note that one of the reason updating stops is that there is lack of tools which enable this. (Kruschitz & Hitz, 2010)

Challenge which choosing web application as delivery platform creates is the integration of the use of these patterns and their usage to designers or developers workflow. Users needs in this case can be reasoned to be that he can easily pick all patterns that he wishes to use for the design but on the other hand is presented ready made pattern combinations that he can pick and then customize. In order to satisfy the need for prototyping the user would likely want to see how combination of different patterns work together. Designers and developers also have a need to customize the visual designs patterns in order create differentiation (Seffah & Taleb, 2012). This means that either delivery platform should have means to do this customization or it should be easily doable through the code present in the patterns. Users should be able to
download the code that forms based on the combination of the patterns or transfer them for further editing since code presented by the patterns isn't likely to share the same visual aspects as the final design. Furthermore the delivery system should in theory be able to account the fact that code will likely to be changed to work with content management systems and even provide the option to automatically add integration to most popular systems in order to keep users from duplicating their efforts.

First literature review section's one of the main findings of the study's literature reviews are the description of benefits and issues of pattern languages. Seffah and Taleb (2012) argued numerous benefits which were reinforced by the findings of Pan and Stolterman (2013). One example of the key arguments for benefits of patterns by Seffah and Taleb (2012) was the ability to capture best practices with problem and context orientation. Second argument that was pointed also by Pan and Stolterman was the potential to act as tool for communication tool between interdisciplinary teams can be seen as important since it highlights the need to communicate the design to other people involved. Key notion was also that patterns can be presented in a way that allows different implementations to create variations in look and feel of the final design. What was missing in the literature was quantitative measurements and evaluation of effects that pattern utilization brings. Much of arguments presented were based on the opinions and experience of researchers focused on studying patterns and pattern languages.

Numerous issues related to patterns and pattern languages were also discovered. Seffah and Taleb (2012) highlighted the lack of interrelationships between the patterns and lack of standard of documentation for pattern languages as defects in pattern languages. The implemented artifact attempted to bring interrelationships between the patterns by arranging patterns into layouts under site types. Artifact enabled assigning patterns also into categories for enabling finding the patterns more efficiently.

Pan and Stolterman (2013) also found that organization of the patterns is an issue. It could be argued that organization is an issue due lack of interrelationships which could evaluated by further study on the topic. This issue was similarly attempted to be tackled in the artifact by arranging patterns with easily navigable taxonomy and listing related patterns with each layout.

Pan and Stolterman also highlighted the issue that developing pattern languages is challenging due time and sufficient knowledge needed to develop good one. Their prediction was that its unlikely that pattern languages will become successful because they haven't been deployed on larger scale yet which can be interpreted as poor argument since it doesn't address the reasons preventing this from happening. It also could be argued frameworks such Bootstrap are pattern languages and have seen widespread adoption in web development although academic studies of spread haven't been conducted.

Another main topic was forming a design process with pattern languages. Forming process around patterns was key interest of Seffah and Taleb (2012). Multiple workflows were presented in the literature but actual adoption of them was observed to be low and Seffah and Taleb noted that commonly agreed process doesn't exist. Authors argued key requirements for the pattern based design process. First requirement was that process should utilize the current issue of the designer to guide him towards other relevant patterns. In the artifact this is solved by guiding the designer or developer with the taxonomy. User navigates by picking site type and then layout under site type based on his needs. Secondly the process should demonstrate how patterns could be utilized by the designer and illustrate combinations with related patterns according to authors. This also issue which the artifact's design of navigable taxonomy attempts to tackle.
Literature review also presented the importance of user experience knowledge as starting point to developing patterns. Seffah and Taleb (2012) argued even that because pattern languages have neglected to offer information about users and their user experience it has prevented patterns reaching their original promise of offering solutions to usability-specific problems. Welie and Trætteberg (2000) argued that patterns should be presented from the end-user perspective in order to end up with solutions that aim to be beneficial to the actual user of the system instead of its creator. In the artifact this issue is approached by presenting patterns as layouts which include context oriented solutions.

Highlighted part of the requirements was also the importance of the pattern representation. Seffah and Taleb (2012) stressed the importance of actually presenting patterns to their user in a way that they can master the pattern and easily and effectively apply them. Seffah and Taleb argued that patterns could feature expert analysis and comments, examples where solution has been already applied and even code examples. It could be interpreted that supplying well authored code with pattern definition would lead to increase in pattern language adoption due to popularity of frameworks. This is indicated in notion by Lidsley's (2013) argument that web development frameworks have become a source for standardization for web development. In the artifact patterns and layouts offer description and context of use and ability link existing implementations. Code of the patterns and layouts is also downloadable in the artifact implementation of this thesis.

The main findings of the second part of the literature review are focused on the aspects of web environment that can be seen to have an effect on the design of pattern languages but haven't been discussed in pattern literature. Study started this review with evaluation of the responsive design. Findings point out responsive design is found out have benefit of providing unified code base which is seen to reduce development and maintenance costs which is according to Pandey (2013) especially apparent in application building. Responsive design concept was also found to have issues that aren't automatically solved by the paradigm itself. Matsudaira (2013) outlined that performance is of the aspects which importance rises when targeting mobile devices. Pandey (2013) evaluated that responsive design might not be able to deliver best possible user experience for applications that require complex workflows and specific adaptations. Outlook for the potential of the responsive web design was found out to be generally positive and benefits the paradigm brings to pattern language can be seen to outweigh the issues. Based on this research each layout in the artifact is implemented with responsive design.

Study also presented that there are tools and publications outside academic research presenting features of pattern languages in context of web development. Most influential development for the future can be evaluated to be the combination of standards by W3C under the name Web Components. This evaluation is based on the finding of Harper and Chen (2012) that open technologies strive when browser vendors add native support for them. Regarding to adoption of technologies study also presented thought by Lindsley (2013) that is important to ensure that even the novice developers and designers are able understand how to customize the solutions offered by the framework. In the pattern language context this would mean that if language would offer also code base for the development in a way frameworks do then guidance should be offered to developers to utilization of this code also and not just the context of usage. In the implemented artifact customization of layouts is implemented with drag and drop to ensure easy customization. What is lacking in the artifact is the documentation of code is but Bootstrap's documentation can utilized instead.
Study introduced challenges that can occur when users are interpreting the patterns in web environment. Islam's (2013) factors that affect the interpretation of interface signs were presented because they pointed out that users interpretation of the web content can depend combination of multiple factors. Islam also presented the notion that multiple signs can clarify the meaning behind the content. These notions can be seen important for the pattern language design since they point out that position of the pattern, its color and reinforcing elements such as images and icons can change the patterns meaning for the user. When thinking about the interdependence of the patterns its worth pointing out that mentioned factors affect meaning of other patterns that are being utilized. This was addressed in the design of the artifact by bundling patterns into layouts which include position of elements in context, placement of images and usage of icons. Importance of icon usage was further evaluated with literature review in order find out if their usage would provide significant advantage over labels. Literature review eventually suggested that icons should be presented together with labels in order to account for first time users who might need to learn their meaning as concluded by Wiedenbeck (1999). This wasn't addressed specifically in the artifact in other ways than opting to use labels instead of icons in many occasions.

Importance of addressing users cultural background was also discussed with the design science research by Reinecke & Bernstein (2013). Culturally adaptive patterns could be seen as possible development but as noted by the authors the means to identify users cultural background are currently limited when location doesn't reveal necessary the correct background. Authors themselves suggested questionnaire to collect the relevant information which could be predicted be too cumbersome to average website. Culturally adaptive patterns wasn't implemented in the artifact, since it was evaluated that it would have required too much resources to implement.

This paper also evaluated pattern languages potential as a tool for solving accessibility issues in web environment. Harper & Chen (2012) shared the evaluation of Fogli, Parasiliti Provenza, & Bernareggi (2013) that current standards WCAG 2.0 and WAI-ARIA won't see significant adoption due their implementation effort. Harper & Chen (2012) suggested that accessibility issues could solved most efficiently if guidelines would be baked into technical language specifications. Fogli, Parasiliti Provenza, & Bernareggi (2013) proposed pattern language which include bridging patterns to aid the implementation solutions proposed by WCAG 2.0 and WAI-ARIA. Significant contribution was also their list of four main accessibility issues created by modern rich internet applications which could be utilized by future pattern language builders to identify pain points. Based on this review on accessibility it would appear that patterns could offer some improvement in accessibility but inclusion in technical language specifications would be needed for widespread adoption. In the artifact semantic tags are used in the layouts to foster better accessibility implementations. No extensive evaluation for accessibility was implemented which can be seen as limitation in this study.

Evaluation was made in the study of usage of prototyping in the context of web development. Goal was to see how pattern language or pattern languages delivery system could fit into prototyping process. Pandey (2013) argued that mockups should be easily sharable with the client and that client should be able use tools to do so with little learning. Suotome (2013) proposed web-based tool which enables clients directly comment mockup and even adjust them themselves. In order to enable sharing, direct commenting and easy interaction with clients the sharable nature of web would make it a relevant choice to utilize in order to deliver the pattern language to users. In the case of the implemented artifact downloaded layouts can be manually hosted as websites. This enables Interaction and responsive features can to be demoed easily, but manual work involved to such set up. Teixeira et al. (2014) suggested that it could be beneficial
if prototype appears in early stage of development as disposable to communicate to client that changes are easy to make at this stage. In the artifact attempt to address this is with downloaded layouts featuring plain Bootstrap styling which can be used to message that development is in early stage.

Suotome (2013) also stressed importance of sitemaps in the evaluation process. Regarding to patterns it can be visioned that sitemaps could be generated for the client to evaluate automatically with pattern delivery platform that would enable designers to pick their patterns on individual site level. Generation of sitemaps wasn't evaluated as feature that would dramatically enable change in the developers workflow and such wasn't implemented in the artifact. Lastly the second part of the literature review focused on reviewing the role of patterns as persuasive tool for the web. Hasle (2011) and Chu, Deng, & Chuang (2014) argued that it is important to focus on persuasive elements since every website has persuasive aspects in them. Chu, Deng, & Chuang argued also that Persuasive Design concepts help define website's persuasion objectives and purpose. In the artifact this is addressed layouts can include description of persuasive patterns in the layout and theory behind the persuasive rationale.

Chu, Deng, & Chuang (2014) conclude in their study of persuasive triggers effects in online shopping that minimum requirements for establishing positive attitude towards online store includes wide product selection which is an aspect of the user experience that isn't directly necessarily dictated or confirmed by selecting to use certain pattern. It is feature that would need to included in the pattern description and interpreted by the designer. This serves as an example of how patterns themselves might not fulfill their potential unless their context of implementation and users desires are taken into account. As further example of this Chu, Deng, & Chuang point out also the users desire to security and privacy protection in the context of online shopping. Idea presented and argued by Kaptein, Markopoulos, de Ruyter, & Aarts (2010) focused on the thought that ambient technologies could be utilized in order to adapt persuasive techniques based on situation, context and user. It could be visioned that patterns themselves would be able present this sort of awareness in their implementation. Simplified example of this would be a pattern that changes it persuasive technique depending on the device user is accessing the website. In the artifact the only way to simulate adaptivity is that layouts can be customized manually by replacing patterns based on the persuasive information included. Intelligent adaptivity is not implemented in the artifacts patterns.

4.3 User Interface Design of the delivery system

Section focuses on describing the final implemented user interface design and reasoning behind it. Focus is also describing ideal features could be beneficial in produced artifact that weren't yet implemented in final artifact. Original user interface design concepts for the artifact are presented in Appendix B. of the thesis. The design evolved during the building of the artifact to its final form. Evolution happened gradually by testing the interactions as they were built.
4.3.1 Home page

Artifact presents firstly the onboarding process of the users with the introductory homepage. Homepage of the artifact aims to convince the users that there is a reason for them to include this pattern delivery platform and the patterns that it provides in their workflow. Aim is to summarize the value of the product and the patterns in few sentences with enough information to get user to become intrigued enough to explore more of the site.

Home page of the artifact also introduces key navigational elements of the site. Navigation bar at the top enables logged in user to gain access to keystone admin view. Main section of the home page features call to action in form of button that is designed to be the next step that new user should take.

Initial design of the system depicted in the figure 4. envisioned also a footer depicting links to platforms blog, Github page and about page. These were dropped from the final design due not featuring yet any content in platforms evaluation phase. Design of the front page changed more than the other original layouts due going through another concepting phase which aimed to design which would explain both the flow and purpose of the application in easily readable format.
Site types is the page that the user moves when he decides to press the start building button that is highlighted on the home page. Site features instructions for the first time user to get started by browsing the layouts by site types. Navigation indicates to the user
that he is currently seeing all of the site types on the current page but he can switch browsing the patterns by categories that are structured based on the function of the patterns. This taxonomy of patterns is similar to Tidwell's taxonomy (Tidwell, 1997). This page in the artifact also introduces another navigational element breadcrumb. Breadcrumb enables to user to be aware of the path that he has navigated, gives sense of place and enables moving back in the hierarchy.

Main function of this site in the artifact is to enable user to efficiently browse all the provided site types and find the site type that matches the requirements of the web site that the user is intending to design. This is achieved by presenting the site types with rows of thumbnails that feature presentation of the structure of the layout that is typical for the site type. Utilizing thumbnails with presentative images allows users to utilize recognition instead of recall. Original user interface design featured in the appendix envisioned navigation to be possible also by functions of the patterns. Example category from the all functions view would be inputs that would provide link to different patterns that enable users to input data.

Original design of the layout featured in the Appendix B. figure 12 imagined also the possibility to browse patterns based on their functionality instead of what site type or layout the pattern belongs.

4.3.3 Browse layouts view

![Blog layouts](image)

*Figure 6. Browse layouts view of the delivery system.*

Layout view is where the user ends up in the artifact after he has selected a site type. This view's purpose is to allow the user to pick between different layout variations. Different layouts might offer different patterns than the others or offer more functionality by including more patterns. Thumbnails are again utilized to allow user to evaluate different options efficiently. Images of different layouts give user much better description of what are the key characteristics of the layout than what the name of the layout is able to reveal. Development of pattern language revealed also that in the future iterations of the user interface layouts themselves could include categories as in the case of the screenshot would be “Post pages” and “Front pages”.

4.3.4 Customize layout view

Customize layout view is the view where the user is taken after he selects and clicks the layout of his choice in the browser layout view. Purpose of the view is to present all the individual patterns that belong to the layout together. The view is visualized also to support reordering the different pattern blocks and replacing patterns with similar variations. Reordering and adding the patterns happens by drag-and-dropping the blocks in to new order. Each pattern also provides link to specific pattern view which provides information about the pattern itself. Options to remove the pattern and the link to the pattern become visible only on hovering over the pattern in order for the user to be able examine the layout as whole without distractions. Previewing responsive design of the layouts is designed to happen quickly by changing the dimensions of the layout by clicking icons depicting different devices.

Right sidebar in the customize layout view is designed to give information for the designer or developer about the characteristics of the layout and give option to download the HTML and CSS with the layout the user has customized. Characteristics of the layout include the description of the layout which should include description of the unique characteristics of the layout. Sidebar also includes the description of more specific context of use for the layout in order to give users recommendation about when and in what context this layout should be utilized. Sidebar features also possible links to possible websites that are based on the same layout as presented. Kruschitz & Hitz compared different formats of presenting UI pattern and out of those the artifact's solution represents closely to Jennifer Tidwell's form of providing name, sensitizing image, problem statement, when to use, why, how and examples of the pattern (Kruschitz & Hitz, 2009). In the case of this artifact this information is added to also combination patterns forming a layout.

Left sidebar of the view has similar but more directed purpose of presenting the user patterns that are recommended to be used with the layout the user is viewing. New patterns from patterns belonging to same site type can be dragged and dropped to create customized version of the layout. This design choice of adding related patterns to
predefined layouts is attempting to improve on the situation of previous pattern languages where the pattern interrelationships are incomplete and lack the context-orientation (Seffah & Taleb, 2012).

Original interface design featured in figure 8. imagined also release info in the sidebar which would feature info such as release date, author and the version of the layout. Another planned feature was tags section which's purpose was to find similar layouts based on the features present in the layout. Design also featured links to complementing layouts but was replaced with feature to browse complementing patterns to layout. This decision was made to allow user to add new patterns to layout faster.

4.3.5 Pattern view

Pattern view opens is where the user ends up when he clicks link to individual pattern in the customize layout view. Here user can can download individual pattern. Each pattern also has description, context of use, examples of use similarly to layout but this time the they are placed in the context of individual pattern. Possible next course in future development for this view would be to present patterns from same category in the left sidebar. It should be noted that this current design has potential flaw of overemphasizing the interaction with the layouts and thus hiding individual patterns information from the user user would need visit patterns own page see the joined information in the right sidebar.
Admin views are designed using standard KeystoneJS components. Figure 10 illustrates the adding of the new layout to the system. Layouts define the main interconnections in the system as seen in the fields visible in the screenshot. After assigning the name for the layout the user selects what sites list the layout and what patterns listed inside the layout.
Administrating the sites and patterns happens in similar views. Figure 11. illustrates adding new pattern to the system. Important fields visible in the screenshots are the definition of pattern category relationship and the addition of the HTML code of the pattern which will be rendered in the customize layout view.

4.4 Software Architecture

Following section describes the software architecture and technology choices of the developed artifact. Section also describes the models of the system in order provide presentation of relevant relationships between the models and thus provide depiction of interconnectedness.

4.4.1 Underlying technologies

Requirement for the delivery system was rationalized to was to be able to add new patterns with little effort and be able to manage patterns relationships easily. Vision for the software included the possibility of the delivery platform to be adopted by anyone who should desire to develop and redistribute its pattern language such advertising agencies or start-ups. This requirement meant that the learning curve and efficiency of use should satisfy most people able who could be imagined as part of being creating pattern language. This definition might in some cases include for example user experience specialists with limited web development and programming experience.

Modern web applications frequently offer the option to manage and add content which is displayed to the users of the application through administrative view which is most cases referred as content management system. In order to reduce duplicate effort on creating structure and content management specific features an open-source CMS was chosen as basis of the artifact.

KeystoneJS Content management system was chosen as starting point for the development of delivery system. It was chosen as platform based on evaluating its design being ideal for building custom application which featured adding data in predetermined format similar to ability of many CMS's have in creating posts, news or products. KeystoneJS doesn't however limit the developer to using predefined formats,
but is instead designed in fashion that enables developer to create new formats easily. KeystoneJS is also able to automatically generate the required admin view for the formats that the developer defines. Second key ability to easily define relationships between these different types or models that the developer is able to define. KeystoneJS is released under MIT license which makes it suitable also for redistribution as part of the created artifact. KeystoneJS is utilized in the delivery platform for creating key elements for the pattern language in form of key models which form the taxonomy of the language, namely the models of sites, layouts and patterns.

KeystoneJS itself relies its operations on number of existing open-source frameworks. It is built on top of Node.js in javascript language which means the developer should be aware of the event-driven asynchronous nature of Node.js. KeystoneJS build also using the Express.js framework. Express.js could be described as server framework which helps for example setting up the routing and model-view-controller type of structure for applications. It leaves many choices to the developer however such as the choice of object-relational mapping and thus the database type. Developers of KeystoneJS have chosen to use Mongoose framework for object-relational mapping and MongoDB as database. Bootstrap front-end development framework is utilized in KeystoneJS in both as default framework for created site and admin view which the administrative user utilizes for adding or modifying the content of created site.

Model-View-Controller (MVC) pattern is built on top of Express.js framework and designed by KeystoneJS team to fit the context of building custom CMS systems. First authored by Reenskaug in 1979 the software design pattern is utilized many popular web frameworks. At the time of the writing of this thesis its is utilized in front-end frameworks such as Angular.js, Backbone.js and Ember.js. Backend framework examples that utilize the pattern include Ruby on Rails and Django.

Krasner & Pope (1988) outlined in their introduction of Model-View-Controller metaphor that the design would allow to develop applications in a way that pieces developed for one application could be reused in new application and help the conceptual development of application. Describing the practical implementation the authors say that pattern imposes separation between the model of the application domain, the views displaying the state of the model and the control or editing of the views and model.

Syromiatnikov & Weyns (2014) note that MVC used in modern web applications on the server side leads to better-organized code that is easy to maintain and understand due to clear separation of responsibilities. Authors argue that original general principles apply on the Web MVC pattern. They also note differences in contrast to Smalltalk’80 definition of of MVC. Authors argument is that in Web MVC Model only stores data which needs to be displayed on certain view and application logic is triggered by the controller. Syromiatnikov & Weyns also argue that there isn't need for strong direct synchronization between user interface and the domain data in a way Smalltalk’80 MVC implements it. Developed artifact follows the MVC convention used by KeystoneJS. This structure features separate folders and files for models, controllers and views. The design enables multiple controllers to utilize one model and controller passing data from multiple models to the view.

Fielding & Taylor (2002) introduce Representational State Transfer (REST) as architectural style which became the foundation for the modern Web architecture. Authors defined REST as set of architectural constraints which aim to minimize network communication and latency. This style of building applications would then as a result lead to maximizing the scalability and independence of component
implementations. Verborgh et al. (2015) argue that not every website conform to REST constraints by default. They note that disregarding the governing principles can affect the integration and scalability of the web application. Authors list two groups of REST constraints: first being the client-server constraints and second uniform interface constraints. Battle & Benson (2008) argue that there is two basic principles guiding the REST-based design. First principle is that each user-facing component is modelled as a resource. Second principle is that each resource in a web application is identified and resolved through a URL.

KeystoneJS utilizes the developer created List objects to create database entries. The created list objects are utilized both to create database schema with Mongoose and to create necessary elements for the administrative view of the created model. MongoDB
doesn't feature joins to make direct relationships possible, but Mongoose features population function to enable easy to use references to other models.

Views are rendered using the Nunjucks templating language which was chosen for this artifact based on easy to read syntax which resembles HTML. Views also include Javascript code responsible for handling the interactive components on the front-end user side of the delivery system. KeystoneJS supports multiple templating languages which affords flexibility if templating language would need to changed at some point.

Following chapter reviews the relevant models of the delivery system which enable formation of interconnected taxonomy of the patterns. Controllers and views aren't discussed in such detail due not providing new knowledge about structuring of pattern language and following the standard convention featured in the KeystoneJS documentation. Full source code of the developed delivery platform artifact is available from link featured in Appendix A.

4.4.2 Key software component designs

Following description includes the documentation of key models of the delivery system as form of KeystoneJS List objects. Schema defined in following models acts as validation rules for the MongoDB database and at the same time generates matching fields in the admin side of the CMS.

User

User model is the standard model included in KeystoneJS installation. It features data needed to authenticate user in order for him to use administrative interface of the CMS. User model is linked to Site, Layout and Pattern models trough their author field as it enables site users to be searched to be added as pattern author in administrative view.

Site

Site model basic fields needed to present it as highest category for the taxonomy of the pattern language. Title determines the name of the site type. State determines if the site is visible to the users or hidden in draft or archived state. Author field would enable the user to define a person who is responsible for adding the category. Field publishedDate enables the user to manage the date when category was published or updated. Lastly the image field holds the user uploaded image which depicts site category in order to aid navigation.

Layout

Layout model features mostly similar fields as Site model. Two new relationship fields sites and patterns which determine the links and taxonomy between all sites, layouts and patterns. As the user fills sites field with suggested existing sites through the admin view it determines under which sites the layout will be listed. Lastly it features fields for information which is presented along the right side of the layout when user is viewing one. Context field is included to feature context of use for the specific layout. Secondly new field is the examples field which can feature links to example sites which feature
similar layout in real life scenario. Lastly persuasive field generates similar field which can be used to fill with information about persuasive features of the layout.

![Figure 13. Artifact's models their relationships](image)

**Pattern**

Pattern model features many of the fields described in previous models. New field included is the code which presents an code editor field in admin view. User inputs HTML code in to this field which forms the pattern to the client side view with CSS and Javascript code included by importing them as modular files to the web page. New field is patternCategories field which defines the relationship with categories among the patterns which is discussed further in description of Pattern Category model. Order of the fields in the code is reflected to the web admin side which is why patternCategories
field is prioritized to be under the title field.

**Pattern Category**

Pattern Category model is the simplest of the models featuring only the name field. User sets categories with different names such as 'navigation' or 'footer' and patterns even from different layouts or sites can belong to these categories. These pattern categories can be used to group patterns within individual layouts.

4.5 Pattern language

Pattern language created to demonstrate the functionality of the delivery system consists of patterns implemented in HTML and CSS languages. Pre-release version of the Bootstrap framework (version 4) was utilized to provide the CSS styles for the patterns. Patterns that require more complex interactions would eventually require developer to include Javascript based code, but this was excluded from the patterns implemented in scope of this thesis. As noted in literature review section of the thesis using component format such as Web Component format for the patterns would have been another option for structuring the code of the patterns.

Highest category in the pattern language is sites consisting three different site types. These site types then include total of 9 layouts. Evaluated set of patterns consist of total 35 patterns included in layouts. Each of the patterns belongs to one of the 8 created pattern categories and have to possibility to belong to multiple layouts. In practice almost all of the patterns belong to just one layout in order to create enough variability among the demonstrating layouts.

A loose language for naming the patterns was also developed in order to better be able to recognize the structure and the context of the pattern without seeing the full presentation of the pattern. Ability recognize the pattern with just it's name was especially important in the management of the patterns since in the current iteration the system lacks the ability to browse patterns with the aid of their visual presentation or images. This is first iteration of such naming and comes with weaknesses of such implementation, but enables the user to separate different patterns than without it.

Issues that came up related to design of the pattern language were related to management of the code in a reusable manner. Issue was apparent in the case of creating multiple structurally similar navigation patterns which were assigned as different patterns because their text labels contained context specific information. From pattern user point of view this creates more understandable presentation of the context, but creates code management issues from the point of view of the admin. This would become apparent if admin wanted to change the structure of duplicated patterns.

In the future iterations it would recommend to explore the possibility of including way of defining variables into the patterns that would allow the reduced duplication of code and yet context aware texts. Another possible approach would be to use generic texts in the patterns. Example of this approach would be defining for example navigation pattern filled with text “link” in each label for the links. This approach would however lack the contextual meaning when defining each link to be named as something that is customary for the context of the site type or the layout.
As demonstrated in the delivery system artifact presentation each pattern featured fields for description, context of use and possible field of links to example sites using the pattern. Usage of description and context of use is demonstrated in each pattern in the constructed delivery system in order to demonstrate their value.

### Table 1. Layouts of the pattern language

<table>
<thead>
<tr>
<th>Layout name</th>
<th>Site type</th>
<th>Description</th>
<th>Persuasive patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front page: Column article links</td>
<td>Blog</td>
<td>Blog layout with focus on featuring the newest blog posts utilizing full width headtexts with link to post. Alternatively posts can be featured with columns with image and headtext.</td>
<td>-</td>
</tr>
<tr>
<td>Post page: Centered</td>
<td>Blog</td>
<td>Blog layout with focus on displaying the blog article with large header image. Layout includes social icons from Font Awesome for post sharing.</td>
<td>-</td>
</tr>
<tr>
<td>Front page: Columns</td>
<td>Portfolio</td>
<td>Portfolio layout with focus on displaying projects with images and descriptions on their sides. Includes also pattern displaying projects with centered headtext which can present details of project also with background image.</td>
<td>-</td>
</tr>
<tr>
<td>Front page: Thumbnail</td>
<td>Portfolio</td>
<td>Portfolio layout with focus on displaying projects as thumbnail images. Includes example of more extensive footer.</td>
<td>-</td>
</tr>
<tr>
<td>Front page: Big images</td>
<td>Portfolio</td>
<td>Portfolio layout with focus on displaying most important projects with full width images. Includes also pattern displaying projects with smaller two column sets.</td>
<td>Testimonials pattern in this layout could be used to persuade the user through recognition. See description of PSD model for details.</td>
</tr>
<tr>
<td>Social: Leaderboard</td>
<td>Application</td>
<td>Application layout with focus on presenting leaderboard between competing users. Layout includes imagined way to load more users to the list without implementing the functionality.</td>
<td>Patterns in this layout could be used to persuade the user through competition. See description of PSD model for details.</td>
</tr>
<tr>
<td>Social: Recognition</td>
<td>Application</td>
<td>Application layout with focus on presenting links to profiles of other user in form of cards.</td>
<td>Patterns in this layout could be used to persuade the user through recognition. See description of PSD model for details.</td>
</tr>
<tr>
<td>Layout name</td>
<td>Site type</td>
<td>Description</td>
<td>Persuasive patterns</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Social: Newsfeed</td>
<td>Application</td>
<td>Application layout with focus on presenting a stream of content produced by user and ways to interact with the content by commenting, by sharing or by expressing interest in the content by loving/liking it.</td>
<td>Patterns in this layout could be used to persuade the user through social learning, social facilitation and social comparison. See description of PSD model for details.</td>
</tr>
<tr>
<td>Social: Join a group</td>
<td>Application</td>
<td>Application layout with focus on presenting links to groups user can join in form of cards.</td>
<td>Patterns in this layout could be used to persuade the user through cooperation. See description of PSD model for details.</td>
</tr>
</tbody>
</table>

Each layout featured similar fields and integration of persuasive design in the pattern language with a field named “Persuasive patterns”. This field describes briefly if the selected layout features combination of patterns which has intended persuasive effect. In table 1. Persuasive patterns fields are featured in same format as in the implemented delivery platform. Field can include by design references to theories and persuasive design frameworks which explain the theories in detail. In the future implementation of this design it could imagined that the delivery system itself would contain database of common theories which could be linked to pattern in similar manner that patterns now link to each other. In the current implementation link is included to research paper introducing the PSD model.

Selection of designed patterns to this pattern language was guided by the need to demonstrate the interconnections and taxonomy of the patterns in efficient manner. In complete pattern language multitude of site types could be imagined to categorize the patterns but to ensure focus on quality only three were selected for this implementation. Portfolio and blog site types were selected due their popularity and familiarity with the designers and developers. Inclusion of application site type was guided by the decision to include persuasive knowledge to patterns. Analysis of principles of PSD model by Oinas-Kukkonen and Harjumaa (2009) revealed that many of principles are best suited to be used in application context. This was especially apparent for the principles under social support category, which was the category that was picked to guide he design of layouts in the application category. Persuasive pattern was also featured in one of the portfolio layouts to demonstrate implementation persuasive guidance in non-application context. Potential to introducing persuasive guidance for websites was recognized in the implementation, but more apparent implementation of the feature would have required larger amount of patterns to be implemented. Application category was included also as see if special treatment would be in order for the patterns of this category. Implementation revealed that patterns featured in the application category aren't as easily combinable with each other since they are designed to work in certain layout context.

Role of the accessibility with the created patterns was attempted to keep in mind when defining the structure of the patterns by including the most common semantic tags, but extensive performance evaluation for screenreader use due resource limitations. Layouts and patterns themselves could feature field explaining the specific accessibility considerations needed to take in to account.
5. Evaluation

5.1 Evaluation method

Framework for evaluation in Design Science Research by Venable, Pries-Heje, & Baskerville (2012) was utilized as basis for designing the evaluation. Based on the suggestions of the framework the chosen method for evaluation was qualitative survey to expert evaluators based on experiment with the artifact. Tasks and the survey of the evaluation is depicted in the Appendix C. of the thesis.

Experiment was conducted online with screen recording of the usage of the system captured by Validately service. This setup enable the reach to expert evaluators in efficient manner but at the same time allowing to confirm that evaluators are able to use system in manner dictated by the experiment and observe potential issues through their usage. Limitation of this method is that doesn't necessarily enable researcher to catch misunderstandings of the questions as well as interview format does.

Focus of the evaluation was to reach potential representative users of the system meaning in this case people with experience in developing user interfaces for websites. Goal of the pattern delivery system developed isn't enable anyone to build websites but enable developers to have access to reusable theory and practices in a way that improves their workflow. Goal of the evaluation is to find out if the developed artifacts could bring improvement to web developers workflow. Related to this the evaluation attempts to gain knowledge if the claimed improvements in form of to pattern language are desired additions. Secondary goals are to find out what is lacking in the current iteration of the artifact, if the current usability of the system is something that would prevent it's adoption. Artifacts determined to be evaluated are the developed delivery platform and the pattern language. Even though the artifacts are discussed as separate in this thesis they are evaluated as holistic entity since the demonstration of the value of the delivery platform would be hard to evaluate without implemented patterns which are connected by design.

Choice of the evaluation method was driven following analysis. Firstly Artifacts are socio-technical by their nature and their utility can be predicted to vary based on users needs and development and design habits. Secondly properties needed to evaluate are utility and efficiency of the artifacts in context of web development workflow. Thirdly constraints of the research environment are the limited resources available for thesis. Time was a limiting factor for evaluation due development and design of multiple artifacts takes time. In the context of pattern language development it is particularly apparent that development work is never finished due changing conventions. Criticism related to evaluation is that it focuses on gathering opinions of people with experience in developing user interfaces for web sites. Biggest weakness of the conducted evaluation is that it only focuses on the side of the delivery platform artifact that is visible to the user when in fact it should also be evaluated how the platform would work in environment where organization uses it to manage their own pattern library.

Two pilot tests were conducted to gain feedback of the evaluation and figure out if evaluation length was suitable. Pilot testing was needed addition to evaluation procedure and helped modify the tasks and the question in the experiment to more understandable format.
5.2 Evaluation results

Demographics

Participants of the evaluation were selected using purposive expert sampling meaning that participant had previously gained understanding of the problem area. In case of this study the participants had requirement of having been involved in developing user interfaces for websites. Evaluators were reached through existing professional network. Participated evaluators had all at least approximately half a year of experience of involvement with user interfaces websites professionally based on conducted survey. Most of the participants excluding one participant had at least 3 years of experience according to survey. Participants evaluated their general involvement in developing user interfaces for the web to between 1-15 years of experience. In total 6 evaluators participated in the experiment. All of the participants in the experiment were male.

5.2.1 Supporting the artifact's potential

Usability

Usability of the systems implemented features was viewed overall as being on good level based on the responses. Four out six respondents had positive notion in their evaluation of usability. Applicable feedback on application's features was also received.

- “Usability is good and consistent. All the actions that I wanted to do were intuitive and the buttons were located in the right places.”
- “.. - DON'T add too much configuration options, I like how simple the system is – great for lo-fi work”

Centered communication

Evaluators indicated that in the case of pattern library that organizations could maintain themselves the key advantage would be the ability to have consistent language between developers. Five out of six evaluators reacted positively when asked about the if such platform would be needed but each had more requirements make such thing possible.

- “..As a developer I would be interested in maintaining a pattern library with this kind of a tool and then allowing other developers to use this as a reference when building their parts of the app. The consistency is the key. The design decisions should be consistent and with a pattern library it is much easier to stick with that...”
- “..It's important to have one place where to find all the patterns and content types available for your site or app or whatever. It helps to keep things consistent, helps to know what kind of patterns are available. Also it helps the developer when he thinks "i wonder what happens if I change this one thing..." if he can easily scan through all the patterns in the site in one place, and not having to browse the site, looking for some specific page with a specific pattern. But a pattern library should be integrated to the organizations' own site/system to be most useful.”.
• “...Best practises (meaning some sort of guide on how to achieve X) are always a good thing, and with this tool an organization could theoretically create example layouts for different situations. I’m not a 100% sure this is needed however, since you can only plan proactively for the "generic" cases (like say, an online store), and those generic cases are already quite easy to design, since most developers are very familiar with similar sites. Sure, if I was working in a company that specialised in online stores, I would appreciate it if we had ready-made templates for different situations, but I half-expect those companies to be able to create and maintain such templates even without a tool like this. Although I guess it could be useful for them to have their templates in one place, where a developer could take whatever templates he wants and customize them accordingly.”

Pattern presentation

Evaluators were asked on their thoughts what information would they present together with an individual pattern besides the pattern itself. Four of six answers indicated that current information the artifact presented along the pattern was sufficient. One evaluator indicated that only Document Object Model and CSS styles would be needed.

• “I think the icons and names were great, and the summaries on the right side explained the typical use cases well. Maybe explain the layouts BEFORE selecting, so there's no chance of "oh no I chose the wrong one"? ”

• “Instructions, tips and use cases for the pattern. What size images to use etc. ”

• “DOM and styles.”

5.2.2 New requirements for the artifact

Overall reaction from the reaction to tool was that it would not currently fit their workflow very well although 3 of the evaluators indicated that it could be good fit in certain scenario. This category lists the requirements evaluators indicated they would like see the artifact to fulfill.

Prototyping

One of the respondents saw the possibility that developed artifact would be good fit in the prototyping phase of development. Requirement for this to succeed was also discussed in answers as being able to easily share the customized layouts which can be navigated by the prototype testers.

• “This system seems great for fast lo-fi mockups, before pulling Photoshop or Sketch out for more high-fidelity work. Great for coming up with prototypes fast”

• “..Make the design's linkable, clickable prototypes rock..”

Another evaluator indicated that could use the output as example even though in most cases the patterns wouldn't fit to his workflow.
• “I could probably create some pattern I like using this website, download it, and then just use the CSS, maybe the HTML as an example. In fact, in such a case this tool could be very useful.”

Centered communication

Respondent saw the potential of tool acting as way of creating style guide when developing application.

• “..I would be interested in having this tool as a Style guide to have consistent development of an app...”

Integration

Respondent indicated requirement that system would fit better to his workflow if the patterns would be integrated to a CMS in a way that would allow the CMS to generate manageable layouts with the patterns.

• “..Building a static local version of the site first can be helpful sometimes though, so there could be some benefit as a starting point. To be truly useful, this kind of system would have to be integrated into the CMS, to serve as a live/dynamic repository for the components and patterns used in the site. So that it wouldn't be just as a starting point but an easy way for the content managers to add and edit content on the site.”

Programming

Evaluator indicated that they saw the potential flaw of the system being that layouts would be trivial program using existing frameworks. There was also indications that instead of downloading patterns providing raw code would be faster for the development workflow.

• “..these layouts that were presented in this system are some what trivial to build with existing frameworks (such as bootstrap)...

• “Not really, I like to dig in and craft the code myself to make sure there's nothing that isn't needed.”

Usability of the Code

Two evaluators pointed out the that code used in patterns should be high quality. One of the evaluators pressed for semantic code. Second evaluators wish was to platform to be able to message the user about what is the level of quality of the generated code. Third evaluator indicated that templating language could be utilized in the patterns instead of raw HTML.

• “..As a coder I always stress about the markup, so make sure it's "semantic" (buzz-word alert!) and without unnecessary cruft.”

• “The knowledge that the code it produces is really truly something I would consider using. But that may be a bit hard to convey in the application without looking at the code after downloading..”


- “Using some template language (Handlebars etc) to pair the patterns to content.”

**Editing patterns**

Three out of six evaluators mentioned the desire to be able to create custom layouts. Adding own patterns was indicated as desire as was ability creation of layout with blank layout and all available patterns.

- “Adding modifiers to patterns, different themes and small changes which can change a single pattern to look/behave a bit different in different places/contexts.”
- “In addition I would have liked to see more editing features.”
- “Add a blank template, and make it possible to choose whatever component I like. The readymade sites and layouts are useful for beginners, but are too opinionated for skilled developers.”

**Maintainability**

Evaluator indicated that in the case of pattern library that organizations could maintain themselves the key to success is how easy such platform would be to maintain.

- “The need for platform is huge, just today at work we discussed at how much work it is to maintain a style guide / pattern library, so having a central platform with all the buttons, forms, typography, layouts, lists and combinations of these available would be huge. And if it is easily maintainable it would be a success.”

### 5.3 Requirement changes based on the evaluation

This section aims to present the findings of the evaluation in relation to requirement findings in the literature and present interpretations of newly discovered requirements. Table 3. presents the newly identified requirements acquired based on the analysis of the evaluation results. Table 4. presents the requirements from the evaluation which were supported by the evaluation results. Findings are discussed further in discussion section of thesis which follows.

**Table 2. New identified requirements based on the evaluation**

<table>
<thead>
<tr>
<th>Req. No.</th>
<th>Requirement from evaluation</th>
<th>Interpretation of the requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.</td>
<td>Pattern integration as building blocks in content management system</td>
<td>Pattern language delivery systems should be designed to integrate with other systems such as content management systems.</td>
</tr>
<tr>
<td>17.</td>
<td>Usability of the code</td>
<td>Code featured in pattern language should be semantic and high quality code and this should be demonstrated to developers.</td>
</tr>
<tr>
<td>18.</td>
<td>Ease of maintainability</td>
<td>Pattern language delivery systems should easy</td>
</tr>
</tbody>
</table>
Table 3. Evaluation requirements with relationship to requirements found in literature

<table>
<thead>
<tr>
<th>Requirement from evaluation</th>
<th>Requirement in the literature</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototyping</td>
<td>Mockups should be easily sharable with the client and that client should be able use tools to do so with little learning. (Pandey, 2013)</td>
<td>Indication that patterns are potential tool for prototyping purposes was discovered.</td>
</tr>
<tr>
<td>Centered communication</td>
<td>Patterns could act as communication tool between interdisciplinary teams. (Pan and Stolterman, 2013)</td>
<td>Potential that patterns could act as communication tool was supported by evaluation feedback. Potential to provide centered communication place was discovered.</td>
</tr>
<tr>
<td>Pattern presentation</td>
<td>Patterns could feature expert analysis and comments, examples where solution has been already applied and code examples. (Seffah &amp; Taleb, 2012)</td>
<td>Support for featuring description and use cases with patterns was indicated.</td>
</tr>
</tbody>
</table>
6. Findings

This section aims to give summary of all the identified requirements for designing and developing a delivery platform for patterns. Table 4. summarizes both requirements found from the literature and requirements discovered through the evaluation of the artifact. Requirements 1-15 are based on the literature and 16-20 are based on the evaluation.

**Table 4 Summary of identified requirements**

<table>
<thead>
<tr>
<th>Req. No.</th>
<th>Requirement</th>
<th>State of implementation in artifact</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Patterns could act as communication tool between interdisciplinary teams. (Pan and Stolterman, 2013)</td>
<td>Delivery platform provides unified place to access the patterns for the teams. Only design is featured in current version.</td>
</tr>
<tr>
<td>2.</td>
<td>Lack of interrelationships between the patterns. (Seffah &amp; Taleb, 2012)</td>
<td>Attempt to bring interrelationships by arranging patterns into layouts under site types. Patterns also assigned into categories.</td>
</tr>
<tr>
<td>3.</td>
<td>Organization of the patterns is an issue. (Pan &amp; Stolterman, 2013)</td>
<td>Patterns arranged with easily navigable taxonomy and related patterns listed with each layout.</td>
</tr>
<tr>
<td>4.</td>
<td>Utilize the current issue of the designer to guide him towards other relevant patterns. (Seffah &amp; Taleb, 2012)</td>
<td>Designer is guided by the taxonomy. He navigates by picking site type and then layout under site type based on his needs.</td>
</tr>
<tr>
<td>5.</td>
<td>Patterns should be presented from the end-user perspective. (Welie &amp; Trætteberg, 2000)</td>
<td>Patterns are presented as layouts which include context oriented solutions.</td>
</tr>
<tr>
<td>6.</td>
<td>Patterns could feature expert analysis and comments, examples where solution has been already applied and code examples. (Seffah &amp; Taleb, 2012)</td>
<td>Patterns and layouts offer description and context of use and ability link existing implementations. Code of the patterns and layout is downloadable.</td>
</tr>
<tr>
<td>7.</td>
<td>Responsive design is found out have benefit of providing unified code base which is seen to reduce development and maintenance costs. (Pandey, 2013)</td>
<td>Each layout in the artifact is responsive.</td>
</tr>
<tr>
<td>8.</td>
<td>Ensure that even the novice developers and designers are able understand how to customize the solutions offered by the framework. (Lindsley, 2013)</td>
<td>Customization of layouts is implemented with drag and drop. Documentation of code is lacking but Bootstrap's documentation can utilized instead.</td>
</tr>
<tr>
<td>9.</td>
<td>Multiple signs can clarify the meaning behind the content. (Islam, 2013)</td>
<td>Layouts include position of elements in context, placement of images and usage of icons.</td>
</tr>
<tr>
<td>Req. No.</td>
<td>Requirement</td>
<td>State of implementation in artifact</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>11.</td>
<td>Pattern based solution for accessibility issues in web environment. (Fogli et. al., 2013)</td>
<td>Semantic tags used in the layouts. No extensive evaluation implemented.</td>
</tr>
<tr>
<td>12.</td>
<td>Mockups should be easily sharable with the client and that client should be able use tools to do so with little learning. (Pandey, 2013)</td>
<td>Downloaded layouts can be manually hosted as websites. Interaction and responsive features can this way be demoed, but manual work involved.</td>
</tr>
<tr>
<td>13.</td>
<td>If prototype appears in early stage of development as disposable to communicate to client that changes are easy to make at this stage. (Teixeira et. al., 2014)</td>
<td>Downloaded layouts feature plain Bootstrap styling which can be used to message that development is in early stage.</td>
</tr>
<tr>
<td>14.</td>
<td>Focus on persuasive elements is important since every website has persuasive aspects in them. Hasle (2011) &amp; Chu, Deng, &amp; Chuang (2014)</td>
<td>Layouts can include description of persuasive patterns in the layout and theory behind the persuasive rationale.</td>
</tr>
<tr>
<td>15.</td>
<td>Patterns could adapt their persuasiveness based on situation, context and user. Derived from Kaptein et. al. (2010), Chu, Deng, &amp; Chuang (2014)</td>
<td>Layouts can be customized manually by replacing patterns. Intelligent adaptivity is not implemented.</td>
</tr>
<tr>
<td>16.</td>
<td>Pattern language delivery systems should be designed to integrate with other systems such as content management systems.</td>
<td>System doesn't integrate with content management system. Patterns are inserted to a content management system but they can't be utilized to create editable content.</td>
</tr>
<tr>
<td>17.</td>
<td>Code featured in pattern language should be semantic and high quality code and this should be demonstrated to developers.</td>
<td>Code quality of the patterns can be only viewed after downloading the patterns. Semantic tags and structure of the patterns could be further improved.</td>
</tr>
<tr>
<td>18.</td>
<td>Pattern language delivery systems should easy to maintain.</td>
<td>Effort is needed to manage patterns due need to manually add images to patterns. Effort is also needed to manage similar patterns with differing context of use.</td>
</tr>
<tr>
<td>19.</td>
<td>Pattern language delivery systems should integrate with programming workflow.</td>
<td>Integration to programming workflow could be improved. Copying code from the site in addition to downloading could be implemented.</td>
</tr>
<tr>
<td>20.</td>
<td>Pattern language delivery systems should allow the user to achieve similar level of control over the results as front-end frameworks offer.</td>
<td>System doesn't allow customizing patterns or constructing new custom patterns.</td>
</tr>
</tbody>
</table>
7. Discussion

The main research question how should the pattern delivery platform and pattern language be build by accounting the previous shortcomings of pattern languages found in the literature. Hypothesis was that if improvements suggested by the literature were implemented in to format that enables efficient delivery the result would be pattern language that designers and developers would want to adopt as part of their workflow. Designers could use the artifact as tool for prototyping and developers could utilize predefined solutions as basis of their development.

Findings on how this implementation succeeded can be drawn based on the conducted evaluation. Evaluation indicated that usability of artifact was on high level but its suitability to developers workflow was poor. Seffah and Taleb (2012) argued that it would be useful if comments, context of use and code examples could be presented with patterns. Evaluators were mostly satisfied with information presented together with the patterns which indicated that it could be useful in the future implementations of pattern languages. Implemented layouts and patterns were fairly well-established so it also could be that more novel patterns would be needed to introduced to get warmer response to extra information such context of use and persuasive knowledge.

It was recognized based on the evaluation feedback that changes would be needed to platform in order fit the developers workflow better. Because of this interpretation in the current state the artifact doesn't fully answer the research question. New requirements for the pattern delivery platforms were identified based on the evaluation which could be utilized to further study the concept. Another requirement finding based on the evaluation was the importance to fitting the user's programming workflow. Suggestion from expert evaluators to solve this requirement included embedding the code of the patterns to web site for easy access. More specifically the issue for the users was that frameworks can enable the developer to build the layout easily. Literature on interface design patterns didn't focus on this on practical level.

Evaluation revealed also the requirement for high level of usability of the code. In practice this suggested by the evaluators to mean semantic and high quality code. Based on the evaluation requirement was identified for integrating the patterns to content management workflow which was another aspect that wasn't discussed in reviewed literature. Desire to editing patterns was recognized as requirement from the evaluation. Desires that were featured in the survey results were adding modifiers to patterns, themes and contextual changes to patterns and ability to create layouts from scratch. One of the evaluators saw the system as potential prototyping tool which indicated that instead of using the system as tool for developers the correct focus group could be designers. Pandey discussed (2013) that prototyping tool should enable easy sharable with the client which was direction suggested in evaluation for the tool. Evaluation also reinforced the potential of patterns acting as communication tool between the developers. Pan and Stolterman (2013) and Seffah and Taleb (2012) recognized patterns as potential tool for communication tool between interdisciplinary teams. Evaluation pointed that patterns have the potential to act as unified guideline between the developers and provide common set of practices for the team.
8. Conclusions

The study presented firstly literature review looking at the potential and the requirements of the user interface design patterns and pattern languages in the context of the web development. First section of the study focused on the merits and the potential of the pattern's in the context of HCI in general. Second section of the literature view summarized key findings of the challenges and potential of the patterns in the context of the web. This literature research presented suite of existing solutions and unsolved problems which to base the design of the artifacts. First artifact implementation was the delivery system. After the basic functionality of the delivery system was implemented a pattern language based on system was designed and implemented with HTML and CSS based patterns. The delivery platform and patterns could be argued forming the final pattern language in combination. The resulted combined artifact was evaluated qualitative survey to expert evaluators based on remote experiment with the artifact. Evaluators interaction with the artifact was observed with through recorded screencasts. Research method utilized was DSR methodology with entering point being the problem identification from which the study followed sequential model. Limitation in the research process was that evaluation in quick cycles wasn't employed. This was partly due the design of the delivery system depended much on finalizing the pattern language. Feedback on the initial design of the delivery system artifact was initially gathered but not constantly during the process.

Limitation in the conducted evaluation was that platform wasn't advertised as tool that they can use to add their own patterns. Instead the focus was on finding out if the delivery platform would fit in to their workflow and what weaknesses platform has when compared not using ready made patterns. Evaluation didn't focus on evaluating the persuasive aspect of the patterns specifically. One of the experiment questions was designed with intent to see if persuasive knowledge provided by the artifact would be discussed. No discussion was however stirred in the survey so further experiments would be needed in this are in order to confirm new findings. More suitable sampling method for the would have also been snowball sampling to further eliminate possible bias. Some of the troubles with interpreting the survey questions didn't come up yet in the pilot studies. Even though the evaluators expressed some difficulty in understanding one question the answers were related to correct phenomenon. Usage of interview as surveying method would have eliminated this issue. Limitation in the execution of the study was also that constructed artifact wasn't yet released to the public which means eventual adoption of pattern language cannot be evaluated based on this research. Research instead gives direction which can be further developed.

One of the limitations of the study's literature review is that it is able to only capture only limited overview of all things that affect patterns and what patterns affect. Patterns by their nature of being defined as a tool to capture knowledge they can present knowledge from large scale. Study also relies much on the practical and cumulative knowledge of pattern authors and while knowledge has been evaluated scientifically as pointed out by literature these argued pattern languages haven't been adopted on a larger scale.

Literature review of the study could be extended to also include evaluation of how patterns would work in practice as part of the process of creation of website. Evaluation could also extend to the a have look at role of the pattern during the whole lifetime of
the website. Overview provided in this study didn't focus on evaluating role and integration of the patterns as part of content management process. It could be evaluated if patterns can be directly integrate as part of the content management and adapt to the content input by the user. Limitation of the research was that efficiency and the usability of the admin side of the developed system wasn't evaluated. This was partly to tighten to focus of this thesis to how delivery system would affect the workflow of designers and developers. Partly this was due recognition that developed delivery system had higher learning curve and required still manual work from the pattern authors. Next step in to development and evaluation of the artifact would be to reduce the amount of effort from the part of the author of the patterns.

Based on the findings from the literature goal was to build pattern language delivery platform which would feature interconnections between the patterns and feature additional knowledge about context, description and persuasive features of the patterns. In addition the significant implemented requirement in the system is that user is guided to relevant pattern selection by taxonomy based navigation of the delivery platform. Study described the process and design decisions behind building the delivery system and pattern language. Implemented artifact is tied closely to the utilized frameworks. KeystoneJS content management system was utilized as basis of the delivery platform and Bootstrap front-end development framework as basis for the pattern language. These frameworks provided practically validated basis for building the delivery system and the pattern language.

Findings discovered during designing and iterating the delivery platform included the experience that application patterns require different way of displaying interconnections when their patterns can only be utilized in certain contexts. Process also revealed the need to group patterns themselves in to categories which are not tied to layouts but instead their functionality.

Results of the study includes list of possible requirements and potential improvement possibilities for the interaction design pattern languages for the web development derived from literature. Artifact development generated also new knowledge about the issue field. As the development of the artifact progressed the understanding of the issues in pattern creation became more pronounced. Major implication for the idea that organizations could create their own pattern library based on the system was that creator had to create also images depicting the pattern and manage multiple variations of the pattern. Creating image presenting the each pattern is time consuming process and would need to be solved in the future. One way to solve this issue would be to generate screenshots of the patterns automatically and possibly allow the user to adjust the results. Selection of requirements and possible improvements were implemented as resulted artifact. Result of study includes the description of creation of the artifact and results of it's evaluation. Source code of the delivery platform is also available which could be used starting point for future research. Implications that could be argued of the in light of the results is that pattern languages have untapped potential in web technology based user interfaces. Evaluation indicated that there is a need for organizations to maintain own pattern libraries in order to enable better communication between team members. This study didn't yet provide definitive answer to how to directly unlock potential of pattern delivery platforms, but instead provided set of new requirements to build upon. Constructed artifact presented new way of finding patterns based on problem of the builder and new ways to interconnect the pattern using categories. Novelty of the artifact is also is that it enables presenting additional information along side the patterns and layouts in form of description, context, examples and persuasive knowledge. Previous academic pattern languages focused only on providing the designs for the patterns whereas build artifact delivered usable code for the user in layouts which give context for the user.
Literature review revealed multiple research possibilities that were yet left out of the scope of the artifact build in this thesis. Patterns were suggested as possible solution for dealing with the issue of people with different cultural backgrounds having different reactions to user interfaces. They were also suggestion in the literature that patterns could act as delivery method for improving accessibility. The study also didn't yet focus on delivering patterns as single components which would also possibly include functionality. Evaluation of the artifact revealed future research possibilities of investigating how pattern libraries should be integrated to CMS systems. Interpretation of findings resulted also as requirement to produce semantic and high quality code. More evaluation and development should also be conducted towards how to make pattern delivery systems more easily maintainable and delivery systems integration to programming workflow. Development directions for the delivery system include based on the evaluation interpretations the direction of prototyping tool, integration with content management systems and thirdly tool which would enable developers to replace programming with front-end frameworks altogether. Several systems aiming to similar outcomes have been developed in recent years outside academic research. Evaluation of their weaknesses and strengths could present important knowledge for the development of web development practice.

Literature suggested also the possibility that HCI patterns will take power a way from designers and put into the hands of the users. The evaluation also presented thoughts how patterns could be tool for the content manager and not the developers. Next big leap in this area of research could be predicted to be to figure out best ways to utilize patterns without the involvement of any designers or developers.
References


doi:10.1007/978-3-540-70816-2_8


doi:10.1109/ComputationWorld.2009.45


doi:10.1007/978-3-642-39200-9_47


Appendix A. Repository links & Demo

Repository for the delivery system: 
https://github.com/samnes/sitepattern

Temporarily accessible demo of the system:  
http://efc6e1fa.ngrok.io/
Appendix B. Initial interface design

**Figure 14.** Site types view of initial interface design.

**Figure 15.** Browse layouts view of initial interface design
**Figure 16.** Pattern view of initial interface design.
Appendix C. Evaluation experiment tasks and questions

Tasks

Task 1: Familiarize yourself to website's landing page. - When you are finished, move to the next task from this bubble - regardless of whether or not you were able to perform the task.

Task 2: Navigate to a layout which is under "Blog" site type.

Task 3: Add more patterns to the layout and then download it.

Task 4: Navigate to a layout which is under "Portfolio" site type.

Task 5: Add more patterns to the layout and then download it.

Task 6: Download a layout from under Application site type.

Task 7: Finally familiarize yourself freely to the application or the downloads. - Survey consisting of 8 questions will be available from this bubble after the task by clicking done.

Qualitative Survey

1. How long have you been involved in developing user interfaces for websites? Specify your experience in years.

2. Describe briefly what kind of experience do you have in developing user interfaces for websites?

3. How would the system fit to your current workflow of developing websites or web applications?

4. What information would you present together with an individual pattern besides the pattern itself?

5. How would you improve the system’s usability?

6. What could be improved in the system to make it a more suitable tool for web development?

7. Can you describe what benefits the system has over other similar systems you are familiar with?

8. Do you think there is a need for a platform that enables organizations to setup their own pattern libraries? Explain why.