An Argumentation-based Design Rationale Approach to Reflective Design of Persuasive Systems

University of Oulu
Department of Information Processing
Science
Master’s Thesis
Alireza Haghighatkhah
09.11.2013
Abstract

Technology has significantly influenced human lifestyle, industrial sector, businesses and the way people interact with each other. Evidently, information technology is no longer a neutral tool. Intentionally or otherwise, system designers and developers influence our attitudes and/or behaviors for varying motives including promotion of socially valued activities. However, promoting moral values using technologies raises several ethical concerns. For example, what might be ethical for one person could be completely unethical from the other. Moral justification of persuasive systems requires involving all stakeholders in a rationale and open argumentation process. The moral issues are difficult to perceive and comprehend and more importantly, the entire process involves different stakeholders with conflicting viewpoints. Therefore, it is challenging to find an optimal solution to satisfy all stakeholders and argumentation process encompasses both cognitive and social complexities.

In an attempt to address the problem; this study outlines an argumentation-based design rationale application. The proposed solution is to mitigate social and cognitive complexities pertaining to moral justification process. The main aim of this thesis is to study construction and evaluation of the said solution in order to facilitate reflective design and evaluation of persuasive information systems. This study follows formal problem-centric approach and was conducted by applying Design Science Research (DSR) methodology. We carried out an experiment to examine socio-technical aspects of the designed artifact focusing on system usefulness and its effect on the problem context. System usefulness of the artifact was studied by analyzing perceived usefulness and usability of the solution from participants' opinions.

The intended implication of this study was to examine that argumentation-based design rationale could potentially decrease complexity of argumentation process. Likewise, it could lead to improved communication and reasoning between stakeholders. These aspects do not guarantee design and development of information systems that are morally justifiable by all standards. However, it is suggested that there are certain prerequisites that are needed for moral justification of persuasive information systems. From a practical perspective, we developed and evaluated a discourse-supported application that effectively establishes rationale and autonomous communication process. Theoretically, we acknowledge that three major aspects of argumentation-based design rationale applications including communication, reasoning and problem comprehension capabilities have a significant influence on mitigating social and cognitive complexities of ethical reflection relating to persuasive information systems.

Keywords
Persuasive System, Reflective Design, Design Science Research, Design Rationale
Foreword

I would like to express my utmost gratitude to my supervisor, Professor Harri Oinas-Kukkonen for his continuous support and guidance throughout my research work. I would also like to thank the OASIS research group and its members for helping me to finish this project.

I am grateful to the University of Oulu and Department of Information Processing Science for providing me an opportunity to conduct my master study. Lastly, I would like to thank my family for their endless love and support.

Alireza Haghighatkhah

Oulu, November 09, 2013.
## Contents

Abstract .................................................................................................................................................. 2
Foreword ................................................................................................................................................ 3
Contents ............................................................................................................................................... 4
1. Introduction ...................................................................................................................................... 5
2. Background ...................................................................................................................................... 7
   2.1 An Overview of Persuasive Technology ....................................................................................... 7
   2.2 Ethical Challenges and Dilemmas ................................................................................................. 9
   2.3 Approaches Towards Ethical Persuasive Systems ........................................................................ 10
      2.3.1 Guideline-based approaches ............................................................................................... 10
      2.3.2 Stakeholder analysis approaches ....................................................................................... 11
      2.3.3 User involvement approaches ........................................................................................... 12
   2.4 Design Rationale ....................................................................................................................... 14
      2.4.1 Motivation for Design Rationale ......................................................................................... 14
      2.4.2 Design Rationale Explained ............................................................................................... 15
      2.4.3 Design Rationale Approaches ........................................................................................... 16
      2.4.4 Argumentation-Based Design Rationale ............................................................................. 17
3. Research Method ............................................................................................................................ 20
   3.1 Design Science and Information System Research ................................................................... 20
   3.2 Design Science Explained .......................................................................................................... 21
   3.3 Design Science Research Methodology (DSRM) ..................................................................... 22
   3.4 Adopted Research Methodology .............................................................................................. 23
4. The Software Artifact ...................................................................................................................... 26
   4.1 Motivation and Vision ............................................................................................................... 26
   4.2 The Rationale Browser .............................................................................................................. 26
   4.3 The Development Process ....................................................................................................... 28
      4.3.1 Requirement Specification .................................................................................................. 29
      4.3.2 System Architecture ......................................................................................................... 30
      4.3.3 System Design .................................................................................................................. 32
      4.3.4 Artifact Realization and Demonstration ............................................................................. 42
5. Evaluation ....................................................................................................................................... 52
   5.1 Research Setting ....................................................................................................................... 52
      5.1.1 Experiment Objectives ...................................................................................................... 52
      5.1.2 Experiment Design and Implementation ............................................................................. 53
      5.1.3 Evaluation Method .......................................................................................................... 54
   5.2 Evaluation of the Rationale Browser ...................................................................................... 55
      5.2.1 Utility Assessment ............................................................................................................. 55
      5.2.2 Usability Assessment ......................................................................................................... 57
      5.2.3 Open Issues ..................................................................................................................... 59
   5.3 Summary .................................................................................................................................. 60
6. Discussion ....................................................................................................................................... 61
7. Conclusion ...................................................................................................................................... 66
References .......................................................................................................................................... 68
1. Introduction

Information technology (IT) has changed modern societies and their contexts in a number of ways. It has transformed industry; businesses and the way people communicate and interact with each other. The fact of the matter is that, information technology is no longer a neutral tool (Oinas-Kukkonen & Harjumaa, 2009). System developers influence our attitudes and/or behaviors for varying intentions including promoting socially valued behaviors like healthier lifestyle, promoting formation of a safe society, environmental conservation and so on (Oinas-Kukkonen & Harjumaa, 2009; Yetim, 2011b). Aforesaid are a few of voluminous examples of persuasive technologies, which are intentionally designed to change attitudes and/or behaviors through persuasion without use of coercion or deception (Fogg, 2002). Yet, promoting moral goals via technologies raises several ethical concerns (Spahn, 2012).

To address the problem, several methodological frameworks including guideline-based, stakeholder analysis in addition to user-involvement approaches have been proposed. The guideline-based approach provides general guidance on ethical considerations of persuasive systems whereas; stakeholder analysis approach suggests evaluation of direct and indirect stakeholders during the design process without essentially involving them. On the contrary, user involvement approach attempts to involve users in the design process and utilize time for persuasive system in order to democratize ethical evaluation of the systems (Karpipinen & Oinas-Kukkonen, 2013). However, moral justification is possible only by involving all concerned parties in a rationale and open argumentation process (Habermas, 1990). In a way, moral justification of persuasive systems could be seen as an argumentation process in which stakeholders reflect their viewpoints and beliefs relating to system’s conjectures, actions, goals and underlying implicit or explicit values (Yetim, 2011b). The argumentation process involves both reflection of viewpoints as well as rationale judgments of resolutions, actions, goals and values.

From this position, moral justification of persuasive systems includes both cognitive and social complexities. This means that the problem is complex, difficult to perceive and comprehend. More importantly, the problem domain involves several stakeholders with conflicting views and it is therefore challenging to find an optimal solution to satisfy them all. According to Rittel (1972), problem context and solution situation of wicked problems follows the same characteristics as above that make them unique, difficult to comprehend and resolve. Such problems cannot be solved by formal methodologies for example, sequential design process models; instead argumentative approach is promoted (Rittel, 1972; Shum & Hammond, 1994).

Design Rationale (DR) has a long tradition in dealing with wicked-problems e.g. design process difficulties. Originally it aimed to make the design process more transparent, argumentative and explicit for stakeholders (Goldkuhl & Röstlinger, 2009). The main promise of argumentation-based design rationale systems is to provide stakeholders with a communication vehicle to support rational judgment of resolutions (Goldkuhl & Röstlinger, 2009). It supports both the communication and reasoning behind the design process (MacLean, Young, Bellotti, and Moran, 1991). Additionally it could enhance both systems quality as well as quality of the entire process (Oinas-Kukkonen, 1996a).
Keeping the above-mentioned in mind, design rationale systems appear to be a promising approach for facilitating ethical design and evaluation of persuasive systems. However, it is still not clear how and to what extent argumentation-based design rationale applications could facilitate ethical conversation relating to persuasive systems. The main objective of this thesis is to study construction and evaluation of an argumentation-based design rationale application with aim to accelerate ethical design and evaluation of persuasive systems. We were interested to examine in what manners and to what extent proposed solution could truly decrease complexity and improve quality of argumentation process concerning moral justification of persuasive systems.

This study was conducted by applying Design Science Research (DSR) methodology, which can be defined as an iterative process involving construction and evaluation of functional IT artifacts designed to provide an answer to a particular problem. Moreover, Design Science Research guidelines presented by Hevner et al. (2004) and Design Science Research Methodology (DSRM) framework proposed by Peffers et al. (2007) were also utilized in conducting this research.

The main contribution of this thesis is twofold. From a practical perspective, we propose a discourse-support application that effectively establishes a rationale and autonomous communication bridge between the stakeholders. The solution is aimed to mitigate social and cognitive complexities of moral justification process and to improve quality of communication and reasoning among stakeholders. Theoretically, we acknowledge that three major aspects of argumentation-based design rationale applications including communication, reasoning and problem comprehension capabilities have a significant influence on mitigating social and cognitive complexities of ethical reflection relating to persuasive information systems.

This thesis is structured in seven chapters. Subsequent to the introduction, the prior studies related to the moral evaluation of persuasive system and design rationale theory are presented in chapter 2. Chapter 3 elaborates the research method adopted to conduct this study. Chapter 4 describes the development process of proposed solution; chapter 5 explains the evaluation process along with our findings. Chapter 6 focuses on discussion of our results and finally in chapter 7, we have presented conclusive remarks.
2. Background

In this chapter, the theoretical background of the study has been summarized. The first section presents an overview of persuasive systems while; the second chapter focuses on the ethical considerations of persuasive information systems along with its challenges. In addition, prior frameworks and viewpoints concerning ethical consideration of persuasive systems as well as their pros and cons are presented. Finally, the last section discusses the design rationale concepts, approaches and methods.

2.1 An Overview of Persuasive Technology

Persuasion has always been essential part of human interaction. For times people and mass media attempted to influence attitudes and behaviors in a way or the other. Every day we are bombarded with persuasive messages on billboards, television and newspapers; all trying to influence our attitudes and behaviors to purchase a specific product or service, support particular political party or maintain a healthier life-style (IJsselsteijn et al., 2006). Thereinafter, the appearance of modern technologies like Internet and ubiquitous computing have changed the landscape of persuasion and provided us with an opportunity to influence what people think and how they behave (Oinas-Kukkonen, 2013). They have many advantages over traditional channels and for instance they can support two-way communication and provide users with an interactive environment to make more influential experience (Fogg, 2002; Fogg, 1999).

In simple words, persuasion could be seen as a communication process in which a persuader sends a persuasive message to a target audience with an aim to influence their attitudes, behaviors or both without using coercion or deception (Morreale et al., 2001). Persuasion can be categorized into three different types i.e. interpersonal persuasion, computer-mediated persuasion and human-computer persuasion (Harjumaa & Oinas-Kukkonen, 2007). Interpersonal persuasion occurs when two or more people interact with each other via interpersonal communications (Wilson, 2003), computer-mediated persuasion occurs when people try to persuade a specific target audience via computer technology mediums such as emails, instant messages (IMs), blogs and social networking sites. In contrast, human-computer persuasion is more about how computer technologies might affect people and social behavior (Guadagno & Cialdini, 2005).

All of these realizations have led to the investigation of persuasive technologies (PT) (Fogg, 2002; Fogg, 1999). According to Oinas-Kukkonen & Harjumaa (2008), persuasive systems can be defined as “computerized software or information systems designed to reinforce, change or shape attitudes or behaviors or both without using coercion or deception”. It is interesting to note that, PT is not about computer-mediated persuasion; instead it is about studying human-computer persuasion and how it could persuade people attitude and/or behavior in certain situation (Fogg, 2002).

Fogg’s seminal work (2002) explained persuasive technologies and how information technology can act as a tool, medium or a social actor in order to persuade its users. In the past few years, research interest has increased towards systematic study of persuasive systems and various researchers have presented different guidelines, approaches and techniques. Later, conceptual and design frameworks such as Behavior Change Support System (BCSS) (Oinas-Kukkonen, 2013) and Persuasive System
Design (PSD) (Oinas-Kukkonen & Harjumaa, 2009) were suggested in order to provide researchers with a systematic framework to design, develop and evaluate persuasive systems.

More recently, Behavior Change Support System (BCSS) has been considered as an object of study within the research field of persuasive technology. Oinas-Kukkonen (2013) defined BCSS as a socio-technical information systems designed to form, alter or reinforce attitudes, behaviors or an act of complying without using coercion or deception. For illustration, reinforcement means reinforcing of current attitude or behavior in order make them more resilient to change. Altering means changing a user’s response to a specific issue/context whereas, formation occurs when a system attempts to shape new behavioral pattern for a certain state (Oinas-Kukkonen, 2013).

According to the Persuasive System Design (PSD) model (Oinas-Kukkonen & Harjumaa, 2009), development of persuasive system involves three major steps i.e. understanding of key issues behind persuasive system, analyzing persuasion context to recognize the intent, the event and persuasive strategies relevant to the system and finally design actual system qualities. The PSD model also outlines seven postulates that are fundamental issues for all BCSSs. The seven postulates are as follows: (1) Information technology is never neutral but instead it always influences its users in one way or another; (2) People like their view to be organized and consistent; (3) direct and indirect routes are the key persuasion strategies; (4) persuasion is often an incremental process; (5) persuasion through system should always be transparent to its users; (6) persuasion via system must always be unobtrusive to user’s primary task and (7) BCSS should be both useful and easy-to-use to influence its users. (Oinas-Kukkonen & Harjumaa, 2009)

In accordance with the PSD model, the next step is to study and understand persuaders’ role, audiences, messages, channels and persuasion contexts in a larger view. To demonstrate, persuasion context is about the intent, the event and the strategy. The study of intent clarifies role of the persuader as well as target change type whereas, study of event assesses problem domain-dependent features (use context), user dependent features (user context) and technology dependent features (technology context). Furthermore, study of persuasive strategy explains analyses of the message and how the system delivers and adopting a route to reach its users (direct and/or indirect). The final step in designing persuasive systems is actual realization of persuasive software features. It is essential to fully understand information content and software functionalities to develop an effective persuasive system. The PSD model likewise offers wide range of software features and design models for BCSSs under four different categories including primary task support, computer-human dialogue support, perceived system credibility and social support. (Oinas-Kukkonen & Harjumaa, 2009)

Persuasive technologies are a part of everyday life already; they are designed to influence our attitudes and/or behaviors in one way or the other. They are going to be more pervasive, unobtrusive and easily integrated into our lives to promote socially valued activities. As Fogg (2002) points out, persuasive technologies have great potential to be adopted in various areas including health, safety, welfare, education, commerce, environmental conservation and many more.
2.2 Ethical Challenges and Dilemmas

Information technology is no longer a neutral tool (Oinas-Kukkonen & Harjumaa, 2009). Persuasive systems try to influence our attitudes and/or behaviors for different motives; (Oinas-Kukkonen & Harjumaa, 2009; Yetim, 2011b) to promote a healthier lifestyle by increasing physical activities (Consolvo et al., 2006; Jafarainami et al., 2005), to promote environmental sustainability like conserving energy (Petersen et al., 2007) or to stop smoking (Graham et al., 2006).

The major potential of persuasive technology is to help us promote principled targets such as health, sustainability, safety, and welfare etc. However, promoting moral goals via technologies raises many ethical concerns (Spahn, 2012). Numerous critical questions regarding ethical consideration of PT remain unanswered and several researchers have acknowledged it (Fogg, 1998; Törning & Oinas-Kukkonen, 2009). As evidence, in one study all papers published at the first three international conferences on persuasive technology were analyzed and the result showed ethical considerations for PT remained largely unreported (Törning & Oinas-Kukkonen, 2009).

Fogg (2002) defined PT as technologies that are intentionally designed to change behaviors and/or attitudes without using coercion or deception. A similar definition was offered by Oinas-Kukkonen (2013) as definition of BCSS. By having a closer look at these definitions, it becomes clear that users’ voluntary participation in the persuasion process and separation of persuasion from coercion and deception are strongly emphasized (Smids, 2012). Additionally, two principles of seven postulates introduced by Oinas-Kukkonen & Harjumaa (2009) including transparency (P5) and persuasion strategies (P3) have significant implications for the users’ voluntary participation in the persuasion process (Karppinen & Oinas-Kukkonen, 2013).

For illustration, transparency has huge impact on users’ voluntary participation and ethical evaluation of persuasive systems. As Atkinson (2006) and Davis (2009) have outlined, persuasion would be ethical only if users are aware of persuaders’ intent. Yet, Berdichevsky and Neuenschwander (1999) claim that designers of persuasive systems should clarify their motivation and system's intent except when disclosure would considerably weaken an otherwise ethical goal. Moreover, Oinas-Kukkonen (2013) argues that in some situations, systems could influence its users without the user being aware of it (indirect route). However, Smids (2012) claims that “unconscious persuasion” might violate voluntariness condition but lack of awareness does not necessarily mean that persuasive systems are manipulative.

User's deliberate participation and transparency are essential elements of persuasive systems, but they do not guarantee ethical systems (Karppinen & Oinas-Kukkonen, 2013). As Smids (2012) states that transparency and disclosing designer’s motivation and system’s intents are not enough because they do not lead to justifying its meaning. Besides, PT could lead to undesirable outcomes and consequences where designers utilize problematic persuasion method or persuade people in a way that cannot be morally justified (Verbeek, 2006). Apart from all above-mentioned challenges, there are plenty of open questions concerning ethical consideration of PT. Firstly, how we can distinguish persuasion from manipulation (Spahn, 2012). Secondly, how designers with limited perspectives are able to predict expected outcomes (Davis, 2009). Thirdly, it is debatable whether designers can decide which moral values are worth promoting (Yetim, 2011a; Yetim, 2011b). Lastly, it is rather hard to justify morality of a system because ethics and moral values vary from person to person (Gram-Hansen, 2009; Yetim, 2011b).
It is evident that ethical considerations for PT are both challenging and under-studied, as many researchers have pointed out ethical considerations as a “grey area” (Gram-Hansen, 2009; Törning & Oinas-Kukkonen, 2009). In response to this issue, various researchers have presented different methodologies, frameworks and viewpoints to comprehend ethical considerations of persuasive technologies. The subsequent section describes different frameworks and viewpoints presented in previous studies.

2.3 Approaches Towards Ethical Persuasive Systems

According to a conceptual ethical framework proposed by Karppinen and Oinas-Kukkonen (2013), all related prior studies relevant to ethical consideration of persuasive technologies could be categorized into three groups including guideline-based, stakeholder analysis and user involvement approaches. The guideline-based approaches provide general guidance on ethical consideration of persuasive systems whereas; stakeholder analysis approaches suggest evaluation of direct and indirect stakeholders during the design process without involving them. In contrast, user involvement approaches attempt to involve users during design process and use time of persuasive systems in order to democratize ethical evaluation of system. (Karppinen & Oinas-Kukkonen, 2013)

2.3.1 Guideline-based approaches

The main promise of guideline-based approaches is to provide designers with a systematic support to consider ethical issues during the design process of persuasive systems. The study of Berdichevsky and Neuenschwander (1999) was one of the first attempts to directly address ethical issues of persuasive systems. According to their work, analyzing an act of persuasion requires identification of designer’s motivation, persuasion methods and approaches as well as assessment of system’s outcome including intended and unintended consequences. Therefore, they proposed principles of persuasive technology design as a systematic framework to help designers to consider ethical issues during design process.

The framework consists of eight principles: (1) the intended outcome of persuasive technology must be considered ethical even without utilizing persuasion techniques or involving technology; (2) the main motivation behind persuasive technology must not be considered unethical even if traditional persuasion mediums were employed; (3) designers of persuasive systems should take responsibility for all predictable outcomes and consequences; (4) designers must respect users’ privacy; (5) sharing users’ personal information with third parties must be carefully examined for privacy concerns; (6) designers must clarify their motivation, methods and intended outcome except when disclosure would considerably weaken an otherwise ethical goal; (7) systems must not misinform users to achieve its goals and finally (8) systems must not persuade users to do something without their ease. (Berdichevsky & Neuenschwander, 1999)

In another study, Smids (2012) claims that users’ voluntary participation in the persuasion process and her voluntary desire for change is the most critical factor for ethical evaluation of persuasive systems. He defines voluntary change as “a change which is freely made, intentional, purposeful, unforced, or the result of a conscious act of the will.” Moreover, he recommends designers of PT to employ voluntariness assessment in order to identify external controlling influences and evaluate users’ voluntary participation in specific contexts. (Smids, 2012)
Spahn (2012) highlights persuasion as an act of communication in which a persuader tries to influence the users via establishing a communicative relation between the designer and users of technology. According to Habermas and Wahrheitstheorien (1973), validity claims of speech acts are dependent on four variables: (1) claim of comprehensibility; (2) truth; (3) and (4) appropriateness and truthfulness. Therefore, if one of these claims were not fulfilled during a speech-act, conversation might lead to misinterpretations and/or misunderstandings (McCarthy, 1978). In this way, Spahn (2012) argues that validity claims of speech acts and discourse ethics as rationalistic ethics could also be applied in the ethical evaluation of persuasive technologies. He also proposes three principles of persuasion: (1) persuasion should be based on prior approval; (2) the aim of persuasion should be to end the persuasion and (3) persuasion should provide users with maximum autonomy (Spahn, 2012).

Conversely, Løgstrup et al. (1997) rejected the possibility of evaluating ethics based on action and consequences and emphasized that ethics must be considered intuitive. In other words, they have defined ethics as intuitive objects than moral rules based on reasons. Based on Løgstrup et al. (1997), Burri Gram-Hansen (2009) argued that every designer must be aware of ethical responsibility of his or her products. Meaning thereby, designers of technology must respond to the ethical demands and should attempt to develop products that would have impact on the users in a way that designers accept ethically. In addition, ethical reflection needs to be integrated in entire design process but ethical responsibility of technological products is completely dependent on ethical awareness of their designers. (Gram-Hansen, 2009)

It can be argued that guideline-based approaches are necessary, but not enough to guarantee development of ethical persuasive systems (Davis, 2009). Moreover, they are fundamentally subjective because what might be considered ethical from designers’ perspective could be seen as completely unethical from the other person’s outlook (Karppinen & Oinas-Kukkonen, 2013). More importantly, it still remains a question whether designers of persuasive technologies with limited perspectives are capable of predicting future consequences and outcomes (Davis, 2009; Gram-Hansen, 2009).

2.3.2 Stakeholder analysis approaches

These approaches aim to provide designers with a systematic framework to distinguish involved stakeholders and determine how the system might influence them. Stakeholder analysis approaches are widely used in policy science, management theory and business ethics during decision making processes. Nevertheless, the main objective is to identify stakeholders, their behavior, intentions, interests, and conflicts and how the decision might affect them. (Brugha & Varvasovszky, 2000; Goodpaster, 1991)

According to Fogg (2002), ethical evaluation of persuasive systems involves examining designer’s intention, persuasion methods, system’s outcome as well as the target audience. In many cases, designers can rely on their intuition to evaluate what is fair and what is not. However, while examining ethics of a complex setting more systematic methods are required. In this sense, Fogg (2002) proposed a stakeholder analysis framework to help designers better assess ethics of persuasive systems, identify stakeholders and determine how the system might affect them. The stakeholder analysis framework contains seven steps as: (1) list all stakeholders including both direct and indirect stakeholders; (2) identify what each stakeholder would gain and; (3) might lose; (4) evaluate which stakeholder has the most to gain and (5) the most to lose; (6) determine ethics by assessing gains and losses in terms of values and finally (7) acknowledge the values and assumptions behind your own analysis. (Fogg, 2002)
On the other hand, some researchers (Davis, 2009; Yetim, 2011a) argue that existing principles and guidelines for ethical evaluation of persuasive systems are not enough, hence designers must look at the broader human computer interaction (HCI) field in order to find an appropriate method to address ethical issues of designing persuasive systems. Considering human values and ethical issues during design and development of technological solution have a long tradition in HCI field. There are plenty of value-centered methodological frameworks (Friedman, 1997; Hirschheim & Klein, 1994; Sellen et al., 2009). Among all of these approaches, Value Sensitive Design (VSD) is the most comprehensive framework for value-centric design of technological solutions (Friedman, 1997).

VSD is a theoretically grounded approach that helps designers to consider human values during the design process. It can be seen as an integrative and iterative approach consisting of conceptual, empirical and technical investigations. For illustration, conceptual investigation comprises of identification of system’s stakeholders (direct and indirect) and examining how they might be socially affected by the technological design. It involves identification of all related stakeholders along with their values, intention, benefits, harms as well as conflicts. Where, the main aim of empirical investigation is to observe, measure and document human responses to technical artifacts in a broader social context. In this sense, wide range of quantitative and qualitative methods used in social science research could be employed for empirical analysis. Lastly, technical investigation focuses on how system’s design and characteristics might support or deter human values. (Friedman et al., 2006)

Furthermore, Friedman et al. (2006) suggested a practical guideline for using VSD. The guideline covers seven principles comprising of (1) identification of value, technology and use context; (2) identification of relevant stakeholders including both direct and indirect stakeholders who might affected by system; (3) identification of benefits and harms relevant to each stakeholder group and (4) mapping them to corresponding values; (5) conducting a conceptual investigation of key values and (6) identification of potential value conflicts and finally (7) integrating value considerations into organizational structure and objectives.

To date, VSD framework and its practices have been successfully applied by various researchers in different domains such as web browser case study (Millett et al., 2001; Friedman et al., 2002, 2005), urban planning system (Borning et al. 2005), and studies concerning privacy in public (Friedman et al. 2006), just to name a few. In fact, it is established for analyses, design and development of value-centered technologies (Yetim, 2011a). Yet, many critics question the practicality of VSD framework with regards to ethical considerations of persuasive systems. For instance, VSD framework does not provide any explicit guidance for engaging a particular context of use or discovering values (Le Dantec et al., 2009).

2.3.3 User involvement approaches

These approaches mainly focus on involving users during design and use time of persuasive systems in order to address ethical issues. To date, user-centered approaches in PT research field are mainly applied for evaluation of usability and system’s effectiveness but not for addressing ethical issues (Yetim, 2011b). In fact, active engagement of users with an aim of ethical consideration would enable them to provide their viewpoints, beliefs, and values and promote social learning and multi-stakeholder negotiation. Likewise, designers would be able to understand users’ moral reactions to the designed technology (Davis, 2009). In this category, two different approaches have
been proposed that could be further classified into two streams including participation in design and conversation in use time (Karppinen & Oinas-Kukkonen, 2013).

Davis (2009) proposed Participatory Design (PD) as an approach to engage users for ethical consideration of persuasive systems. The main promise of participatory design approach is to actively involve all stakeholders during the design process in order to ensure the end product covers their needs and viewpoints (Muller, 2003). In a sense, PD approaches could help designers of persuasive systems to take users’ views into account concerning open ethical issues. In her prior study, she employed VSD framework to identify direct and indirect stakeholders along with their values. Furthermore, Inspiration Card Workshop (Halskov & Dalsgård, 2006) was applied as an approach to promote stakeholders’ participation. Hence, stakeholders are able to present their viewpoints regarding various topics including design potentials, technological options and ethical issues. (Davis, 2010)

However, Yetim (2011b) argues that explicit guidance during participatory process is required in order to enable users to identify system’s motives, methods and outcomes. More specifically, what questions should be asked during design discourse of persuasive system in order to refine systems’ actions, goals and underlying values. Besides, limitations in number of participants, available design knowledge, and unpredictability in future use context and value changes over time, all together might lead to a situation where we are not able to guarantee effectiveness of system in future use situations. (Yetim, 2011b) To address above-mentioned issues, Yetim (2011b) has proposed a systematic framework to facilitate ethical design and evaluation of persuasive system. The framework has been built upon Habermas’s (1990, 1993, 1996) discourse ethics theory and value-based practical reasoning (Atkinson et al., 2006; Walton et al., 2008) in order to provide designers with explicit guidelines to deal with ethical considerations of persuasive systems during design process and use time.

It is interesting to note that, discourse ethics do not provide optimal norms that regulate between different stakeholders’ viewpoints, value orientations and conflicts. Instead it provides a method to find the norms (Yetim, 2011a). As Habermas (1990) has acknowledged, moral norms cannot be justified individually; rather it is required to involve all relevant stakeholders during the argumentation process. In this sense, moral justification of persuasive systems could be seen as an argumentation process in which stakeholders reflect their viewpoints and beliefs concerning system’s assumptions, actions, goals and underlying implicit or explicit values. The argumentation can be initiated from system’s consequences, underlying values or assessments of persuasion goals, methods as well as intended or unintended outcomes in order to justify its morality. Furthermore, arguments from values promote value-based practical reasoning because it enables stakeholders to assess and evaluate reasons and motives behind a given system’s action and its alternatives. (Yetim, 2011b)

To deal with various issues during discourse ethics, Habermas (1993) has proposed three different practical discourses i.e. pragmatic, ethical and moral domain. For illustration, pragmatic discourse aims to seek motives behind rational choices of actions to achieve agreed goals or to rationally assess whether goals are in accord with value preferences. Ethical discourse comes into play when there is an uncertainty or disagreement about objectives and goals; and lastly, moral discourse seeks to identify norms and assess whether values are equally good for all or otherwise. (Habermas, 1993)
Accordingly, Yetim (2011b) highlighted a set of critical questions that guide argumentation during ethical design and evaluation of persuasive systems. From the perspective of persuasive systems, pragmatic discourse includes questions to identify and check whether system goals and actions are in line with values or whether actions are in accordance with systems objective and goals. Whereas, ethical discourse questions aim to identify value preferences and examine whether results of pragmatic discourse are in accord with the preferred values. Finally, moral discourse questions designed to identify norms particularly when value conflicts occurred and evaluate whether the results from pragmatic and ethical discourse are in concurrence with accepted norms. (Yetim, 2011b)

To summarize, above-mentioned discourses facilitate active argumentation concerning identifying actions, goals and values from pragmatic (purposive), ethical (good) and moral (just) perspectives. Proposed framework promotes a transparent and logical ethical conversation during design process and use time of persuasive systems. It is also possible to employ it to critically analyze and evaluate designed technology individually (by an expert) or discursive (in teams). (Yetim, 2011b) The current work focuses on aforesaid critical questions, as will be described later.

2.4 Design Rationale

This section mainly focuses on Design Rationale (DR) and its approaches. The first part explains the motivation and early stages of system development. The second part discusses about DR potentials and benefits while the third section introduces various DR approaches. Last section elaborates on argumentation-based design rationales and its methods.

2.4.1 Motivation for Design Rationale

Information System (IS) design is essentially a complex process. The process typically involves continuous requirement collection, problem domain investigation, evaluating technical design criteria, alternatives and constraints. Furthermore, it involves various stakeholders with different goals, objectives and viewpoints, all looking for a satisfying solution. Various organizational objectives, conflict of interests, technical complications as well as assessing design tradeoffs, all together lead to the fact that IS design process is very challenging and complex. (Tang et al., 2006; Tang et al., 2007) In this sense, apart from technical difficulties, the design process is also involves cognitive and social complexities (J. Conklin, 2005; Goldkuhl & Röstlinger, 2009).

For illustration, the cognitive aspect of design process implies that usually: (1) the problem is not well-defined and it has many underlying issues; (2) it is difficult to perceive and comprehend problem context; (3) there is a non-linear and complex relationship between elements; (4) there is no optimal solution, only satisfactory solutions could be a resolution and (5) finally the solutions might lead to unforeseen consequences. Moreover, social complexity indicates that various stakeholders with different goals, objectives and preferences are involved in the problem domain and it is difficult to find an optimal solution to satisfy all needs and viewpoints. (J. Conklin, 2005; Rittel, 1972; Rittel & Webber, 1973)

Therefore, many design problems including IS design process can be defined as wicked problems. According to Rittel (1972), problem context and solution situation of wicked problems follows the same characteristics as above, which make them unique, difficult to comprehend and resolve. Therefore, such problems cannot be solved by formal
methodologies like sequential design process models; instead of argumentative approach must be considered. (Rittel, 1972; Shum & Hammond, 1994)

Rittel (1972) acknowledged that the design process couldn’t be seen as a well defined and structured problem in which stakeholders attempts to understand the problem context, collect and analyze information and finally propose a solution. In contrast, design process is a continuous argumentation between stakeholders in which they attempt to investigate problem context, present and evaluate various resolutions in order to find a substantial solution. In this sense, design process can be defined as “constant alternation between designing artifacts and augmenting the designs” (Oinas-Kukkonen, 1996b).

Taking design as an argumentative process, Kunz and Rittel (1970) proposed Issue Based Information Systems (IBIS) as an approach to encourage argumentation during the design process. They claimed that argumentative approach to design is the only way to address wicked design problems. Thereinafter, many researchers proposed different approaches, models and methods in order to facilitate capturing and representation of argumentation for the design process. All of these realizations motivated the development of Design Rationale.

2.4.2 Design Rationale Explained

Previously, it has been argued (Conklin, 1989; Floyd, 1993) that software design process is primarily an artifact oriented process in which development team starts with requirement gathering, specification, designs artifact and constructs the final solution. However, design is a continuous alternation between action and reflection like construction and argumentation. (Fischer et al., 1989) The design process contains constant argumentation about the problem context, resolutions, design alternatives and decisions. In this sense, end product of design process is of dual nature containing both the design product and design process. (MacLean et al., 1991)

Design Rationale (DR) (Moran & Carroll, 1996) aims to understand why an artifact has been designed the way it has (Fischer et al., 1992), which might include information about requirement, assumption, alternative solutions and the reasons behind decisions made (Oinas-Kukkonen, 1998). The main aim of DR is to facilitate recording and representation of argumentation and it’s reasoning behind the design process (Moran & Carroll, 1996).

The goal of capturing design rationale is to make decision-making process based on facts rather than opinions (Oinas-Kukkonen, 1998). Likewise, it might lead to a transparent and rational design process and provide better communication between stakeholders (Conklin & Yakemovic, 1991). Furthermore, DR support more constructive moves and makes problem context and resolutions more transparent and rationale (Oinas-Kukkonen, 1996b). It could also be seen as a communication vehicle between stakeholders in the knowledge environment and might enhance both target system quality as well as quality of process (Oinas-Kukkonen, 1996a).

As admitted by several researchers, DR could bring many benefits to its users. Mostow (1985) argues that it might improve documentation, debugging, verification, analysis, modification and automation of design issues. Fischer et al. (1991) suggest that it supports maintenance of an artifact, reuse of design knowledge and facilitate critical reflection during design process. Moreover, MacLean et al. (1991) claim that it facilitates both reasoning and communication during the design process.
Furthermore, several studies report DR applicability and its benefits in different domains including improving quality of software architecture and design process (Tang et al., 2006; Tang et al., 2007), requirement engineering (Bañares-Alcántara & King, 1996; Whelton et al., 2001), improving inspection process of software artifacts (Tervonen & Oinas-Kukkonen, 1996), conflict mitigation (Pena-Mora et al., 1995), enhancing creativity in distributed environment (Ball, et al., 2001; Farooq et al., 2005; Wang et al., 2013) and so on.

2.4.3 Design Rationale Approaches

Design Rationale involves: (1) basic approaches and conceptualizations toward rationalizing the design process, (2) various notation models to capture and represent rationale elements as well as (3) computer-supported tool and applications (Goldkuhl & Röstlinger, 2009). Naturally, the primary requirement of design rationale system is to capture and represent the rationale in a way that supports its promises (Regli, Hu, Atwood, & Sun, 2000). In a way, the main challenge of DR system is how to capture rationale with minimum cost and, to represent rationale to maximize system’s usability (Regli et al. 2000; Shum & Hammond, 1994).

From a holistic viewpoint, approaches to develop design rationale systems could be classified into two major categories i.e. process-oriented and feature-oriented approaches. For illustration, process-oriented systems generally highlight the design rationale as history of design process. This approach is mostly employed in the domains where the problem contexts are complex, difficult to comprehend and resolve (E. J. Conklin & Yakemovic, 1991). Typically, process-oriented approach is signified as argumentative approach because it focuses on argumentation for the design process. (Regli et al., 2000) However, feature-oriented approach is beneficial in domains where the field knowledge is mature and normally design process is routine and rule-based.

Matter of the fact is that feature-oriented systems typically support generation of design rationale based on existing knowledge base (Garcia & Howard, 1992; Myers et al., 1999). In this way, design rationale capturing and representation are more formal while; argumentative approach (process-oriented) frequently uses descriptive or semiformal models. (Regli et al. 2000)

In simple words, feature-oriented approach provides active support to design activities to develop IT artifacts, whereas argumentative approach aims to facilitate capturing of rationale behind the design process. Feature-oriented approach typically focuses on designed artifact and requirement fulfillment, however, argumentative approach emphasizes on the design process (i.e. problem context investigation, resolutions, design tradeoff, who, when and why). Finally, feature-oriented design rational systems are more beneficial when applied in a mature context, while argument-based design rationale systems are suitable for specific contexts where problem and solution are both complex and poorly understood parallel to wicked problems. (Regli et al. 2000)

To conceptualize, capture and represent a design rationale, many researchers have projected diverse notation models. A notation model can be described as a conceptualized model, schema or language of communication to capture and represent design rationale elements. A notation model can be categorized into three types comprising of informal, semi-formal and formal. In contrast to the informal models, formal schemas are more structured and strict in capturing and representing a rationale whereas semiformal models have advantages of both the approaches. Generally, semiformal models are more appropriate for argumentative design rationale approach where, formal schemas are constructive in feature-oriented systems. (Regli et al., 2000)
2.4.4 Argumentation-Based Design Rationale

Argumentation-based design rationale is aimed to make the design process transparent, argumentative and explicit for stakeholders (Goldkuhl & Röstlinger, 2009). Typically, argumentation during the design process contains information about the problem domain investigation, evaluation of resolutions with reasons behind the decision made (Oinas-Kukkonen, 1996a). The main aim of argumentation-based design rationale systems is to provide stakeholders with a communication tool to support rational judgment of resolutions (Goldkuhl & Röstlinger, 2009). It provides both communication and reasoning for the design process (MacLean et al., 1991) and could improve system quality and quality of process (Oinas-Kukkonen, 1996a).

The objective is to conceptualize argumentation process, its structure and elements. Especially, to know what could be made explicit and what could be assumed understood. Such issues have created motivation for development of various design rationale methods. (Jarczyk et al., 1992) The roots of the argumentation-based design rationale can be traced back to Toulmin’s model of argumentation (1969) and IBIS (Issue-based Information System) method (Kunz & Rittel, 1970). The IBIS model was the first attempt to bridge the gap between design and argumentation (Shum & Hammond, 1994).

The IBIS model (see Fig. 1) comprises of three different nodes (issues, positions and arguments) as well as eight different link types (supports, objects-to, replaces, responds-to, generalizes, specializes, questions, and suggested-by). For illustration, each issue could have several positions as candidate answers and each position might have one or more arguments that might support or object-to it. Moreover, link types explain the relationship between elements of argumentation (Kunz & Rittel, 1970).

![Figure 1. Rittel's IBIS (Issue-based Information System) (Kunz & Rittel, 1970)](image-url)

Various design rationale methods have been developed and reported, for example, the Procedural Hierarchy of Issues (PHI) (McCall, 1991), Design Space Analysis (DSA) (MacLean et al., 1991), Decision Representation Language (DRL) (Lee & Lai, 1991). The main difference among these methods is their view about conceptualizations and argumentations structure and more precisely, to what extent information behind design process ought be recorded and represented (Jarczyk et al., 1992).
Yet, despite the great potential of design rationale, the success rate in the industry is occasional (Jarczyk et al., 1992; Regli et al., 2000). Apart from technical capabilities including systems usefulness and usability, notation models as a language of thinking and communication plays a vital role in this area. As highlighted by Jarczyk et al. (1992), design rationale would be usable if it employs easy to use metaphors. Correspondingly, based on observations from various design rationale methods, Oinas-Kukkonen (1996b) proposed Question-Answer-aRgument (QAR) method to simplify the explicit rhetorical structure of design rationale.

QAR method uses natural terms such as questions, answers, arguments and decisions to express the discussion. As shown in Figure 2, nodes, links and hyperdocuments are the basic representations. For illustration, nodes are information containers that contain text, attachment as well as owner information such as authors and creation timestamps. The discussion expressed in three types of nodes i.e. questions, answers and arguments. Each question expresses the design problems that might have one or more answers. The answers are candidate resolutions to the problem/s and they could have one or more arguments that either support or object the argument. Likewise, the final answer as an agreed upon resolution can be marked as a decision. Moreover, recursive relationship between questions enables one to generalize, specialize or replace design problems with another question. In this way, each separate question could be seen as a hierarchical tree, which might contain child questions, relevant answers and arguments. The questions belong to a hyper-document that contain a collection of discussion consisting of nodes and links between them. (Oinas-Kukkonen, 1996b)

Figure 2. Question-Answer-aRgument (QAR) method (Oinas-Kukkonen, 1996b)
To conclude, QAR is a simplified design rationale method that uses debate and/or discussion as a natural metaphor to represent the argumentation. The terminology used in the method is close to human natural language in order to be understandable and easy to use. Moreover, hyperdocuments facilitate growth and evolution of argumentation in more organized manner that enables organizations to effectively categorize and organize design rationale in different domains. (Oinas-Kukkonen, 1996b)

To date, QAR method has been employed in various studies including evaluating its usability during software design process (Oinas-Kukkonen, 1998), enhancing software quality through the inspection process (Tervonen & Oinas-Kukkonen, 1996), assessing its applicability in a computer-aided software and method engineering environment (Oinas-Kukkonen, 1996a) and evaluating its suitability in sense-making of argumentation-based knowledge among web-based communities (Räisänen & Oinas-Kukkonen, 2007). The current study considers QAR method, as will be described later.
3. Research Method

First two sections of this chapter briefly explain the Design Science (DS) research in Information System (IS) discipline. The third section describes Design Science Research Methodology (DSRM) along with its activities and procedures. And final section elaborates the research method adopted to conduct this study.

3.1 Design Science and Information System Research

The main promise of IS in organizations is to enhance their effectiveness and efficiency. Information system capabilities, characteristics of organization, people, development methodologies and the environment, would determine IS achievement in a particular use context. Information system research aims to develop and communicate knowledge in dual manner in order to (1) support development of productive IT applications for organizations and its management and (2) to develop and communicate knowledge concerning to management of information technologies and more importantly use of information technology for managerial and organizational aspects. (Hevner et al. 2004)

Information system research involves both technological and social systems complexities. The main aim of research in IS discipline is to study both technological and social complexities and examine the phenomenon that emerge when the two aspects overlap. (Gregor & Jones, 2007) Natural science research methods are appropriate to study and understand existing and emergent phenomena; yet, they are rather insufficient for the study of wicked problems, which are unique, difficult to comprehend and resolve. In fact, such problems required creative and innovative solutions to be created and evaluated in a particular context. (Chatterjee, 2010)

Research in IS discipline mostly relies on two distinct paradigms comprising of behavioral science and design science (March & Smith, 1995). The behavioral science approach is based on natural science research. It aims to develop and justify theories that predict or explain organizational and social occurrences concerning analyses, design, development and use of information systems. The main goal is to study and understand the relationship between information systems, organizations and people in order to enhance effectiveness and efficiency. In contrast, design science research is fundamentally based on engineering disciplines and science of artifacts (Simon, 1996). It can be defined as a pragmatic research paradigm that aims to create novel and innovative artifacts to solve real-world problems (Simon, 1996). In other words, it involves creation and evaluation of IT artifacts in order to solve a specific organizational problem (Hevner et al., 2004).

While above-mentioned research paradigms have differences in their nature and origin, they are inseparable in IS research. There is a complementary connection between behavioral science and design science research. While behavioral science aims to develop and justify theories that predict or explain organizational and social phenomena, design science research aims to develop innovative ideas, expand and identify undeveloped capabilities of information systems. In other words, behavioral science research aims to provide the truth by justifying theories whereas, design science seeks to enhance its utility in the larger context. (Chatterjee, 2010; Hevner et al., 2004)
3.2 Design Science Explained

Design Science Research can be defined as a process of creation and evaluation of IT artifacts intended to provide an answer to a particular organizational problem. The designed artifacts are both useful in solving the problem and fundamental in understanding the problem context. In other words, DSR is based on the idea that knowledge and understanding of problem context and solution are acquired in the construction, evaluation and application of IT artifacts. (Chatterjee, 2010)

The design process can be seen as both a verb and a noun. This means that the design process of information system contains both a set of activities (iterative process) as well as a resulting product (artifact) (Walls et al., 1992). In this sense, DSR comprises two processes including creation and evaluation as well as four type of designed artifact consisting of constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems) (March & Smith, 1995). They might contain social innovations (Aken, 2004) or lead to new properties of technical, social or information resources (Järvinen, 2007).

The Information System Research Framework proposed by Hevner et al. (2004) clarifies understanding, executing and evaluating IS research concerning both behavioral science and DSR. The framework explains three research cycles that emerge as result of interaction between the environments, IS research and knowledge base. The environment includes people, organization and technology, which all-together expresses the problem space relevant to the research. The problem space defines the business need or problem that requires solutions. The IS research contains both behavioral science (development and justification) as well as design science (construction and evaluation). In fact, behavioral science aims to justify the truth where, design science attempt to examine its utility. In this sense, they are both interacting and enrich each other. Finally, the knowledge base provides raw materials for the IS research. It is composed of methodologies and foundations in various shapes including theories, frameworks, instruments, constructs, models, methods, and instantiations. (Hevner et al. 2004)

The framework also highlights three main research cycles including relevance, rigor and design cycle. For illustration, the relevance cycle initiate the DSR with a problem domain which consisting of both the requirement for the research as well as acceptance criteria for research evaluation. The rigor cycles connect the IS research to the past knowledge to ensure its innovation. Existing knowledge might contain experiences, expertise as well as existing artifacts and processes from both research domain and application domain. Lastly, the design cycle as heart of DSR contains continuous iteration between construction, evaluation and improvement of artifact. (Hevner et al., 2004)

Besides, Hevner et al. (2004) proposed a set of guidelines to shed more light on how DSR must carry on, evaluate and present. The guidelines comprises of seven practices described as follow:
• Guideline 1 (Design as an Artifact): DSR must produce a viable artifact that could be in the form of a construct, model, method or an instantiation.
• Guideline 2 (Problem Relevance): The outcome of DSR as a technology-based solution must meet the requirement of relevant business problem.
• Guideline 3 (Design Evaluation): The artifact must be evaluated carefully to determine its utility, quality, and efficacy in the problem context.
• Guideline 4 (Research Contributions): DSR should lead to tangible and verifiable contributions in the design artifact, foundations and/or methodologies.
• Guideline 5 (Research Rigor): DSR should be built upon a solid foundation of rigorous methods in both building and evaluating the artifact.
• Guideline 6 (Design as a Search Process): The design process should be seen as a constant iteration process between construction, evaluation and improvement of artifact in order to find a satisfying solution to problem.
• Guideline 7 (Communication of Research): The DSR must be presented effectively to satisfy both technology-oriented and management-oriented audiences.

3.3 Design Science Research Methodology (DSRM)

The lack of generally accepted process models for carrying out DSR motivated development of Design Science Research Methodology (DSRM) (Peffers et al., 2007). The framework contains principles, practices and procedures to guide researchers to effectively conduct and present their DSR researches. It was originally developed to meet three fundamental objectives: (1) to provide researchers with a nominal process model for conducting design science research; (2) must be consistent with existing literature and design science principles; and (3) to provide researchers with a mental model for presenting the DSR research. Figure 3 demonstrates the framework with six sequential activities including problem identification and motivation; define objectives of a solution, design and development of the artifact, demonstration of the artifact, its evaluation and communicating the final results. (Peffers et al., 2006; Peffers et al., 2007)
For illustration, the first activity involves problem identification, motivation and justifying the value of a solution in relation to the problem domain. Knowledge relevant to the problem domain as well as importance of its solution is the main resources required for this activity. The second activity determines objective of the solution in relation to the problem domain. Researchers determine objective of the solution either in quantitative or qualitative terms based on the knowledge of the state of problems and current solution support. The third activity involves creation and construction. Knowledge of theory relevant to the solution enables researchers to determine an IT artifact’s functionality, solution architecture and finally developing the actual artifact. The fourth activity might involve use of artifact in experimentation, simulation, case study, proof or other methods in order to determine its efficiency relating to one or more instances of the problem. The fifth activity focuses on evaluation of the artifact and more specifically, how well the artifact meets solution objectives. In a way, the activity involves comparing objectives of the solution to actual observation results. This activity could involve both quantitative and qualitative evaluation methods. However, selection of evaluation method essentially depends on the nature of problem and solution characteristics. The result from this activity could motivate researchers to iterate back to the initial design and development phase in order to improve the effectiveness of the artifact or to continue over to the next step. Lastly, the final activity focuses on communication. The researcher would need to present the problem and its importance, the artifact including its uniqueness, utility and effectiveness to researchers and other relevant audiences. (Peffers et al., 2006; Peffers et al., 2007)

Although the process is structured in a sequential order, the researchers could originate their research at almost any step and move on. As shown in Figure 3, there are four possible entry points for researchers including problem-centered approach, objective centered solution, design and development centered and observing a practical solution. For illustration, problem-centered approach follows a sequential order starting from problem identification and motivation. In this way, the research might originate from observation of the problem or suggested future research from prior studies. However, an objective centered solution grows from the second activity. The motivation for this approach might initiate from industry or research in developing an artifact to deal with a specific problem. Also, an existing artifact that might originate from another research or problem domain might be used as the basis for the design and development-centered approach. Finally, design science research might initiate from observing an existing practical solution. In this sense, the research would start with demonstration activity to the backwards to ensure rigor and validity of the study. (Peffers et al., 2006; Peffers et al., 2007)

### 3.4 Adopted Research Methodology

The DSRM has been employed to carry out this study. Because of the nature of this research, problem-centered approach has been applied to ensure rigor and validity of the study. The study followed the formal sequential procedure initiated from problem identification and motivation. The following diagram (see Fig. 4) explains the problem-centric approach employed in this study and the following section demonstrates the activities conducted during the research process.
Figure 4. Adopted research methodology in this study

Activity 1: Problem identification and motivation

Several questions regarding ethical consideration of persuasive systems remain unanswered and the problem is also acknowledged as a grey area of persuasive technology research field. The problem has been recognized by prior studies and likewise, it has been identified as one of the most critical future research directions (Fogg, 1998; Törning & Oinas-Kukkonen, 2009).

To fill the gap, various solutions have been proposed to facilitate ethical design of persuasive systems. However, moral justification is possible only by involving all stakeholders in a rationale and open argumentation process (Habermas, 1990). In this way, moral justification of persuasive systems can be seen as an argumentation process in which stakeholders reflect their viewpoints and beliefs concerning system’s assumption, action, goal and underlying implicit or explicit values (Yetim, 2011b). The argumentation process involves both reflection of viewpoints as well as rationale judgments of resolutions, actions, goals and values. However, it is still not clear how discursive approach could facilitate moral justification of persuasive systems? What could be the challenges and difficulties of argumentation process? And how computerized information system might decrease complexities of argumentation process and accelerate moral evaluation of persuasive systems?

Activity 2: Objectives of the solution

This study proposes an argumentation-based approach to facilitate addressing ethical issues related to the PS. The solution enables stakeholders to reflect their viewpoints regarding actions, goals and values of persuasive system under evaluation and likewise, it enable rationale judgment of resolutions. The objectives of the solution could be twofold; the solution acts as a communication vehicle between stakeholders and it supports reasoning behind the argumentation. More specifically, the solution would facilitate active argumentation concerning identifying and checking actions, goals and values from pragmatic (purposive), ethical (good) and moral (just) perspective.
**Activity 3: Design and development**

Formal software engineering methods along with best practices have been employed during design and development of the artifact. Prior studies relevant to this research such as designing argumentation-based design rational applications have been also considered during the development process. Moreover, recent technological frameworks, design patterns and three-layered software architecture model have been utilized during the development phase. The artifact’s functionality, solution architecture and detailed development process is rigorously documented in this thesis.

**Activity 4: Demonstration**

Considering the characteristics of the problem and available resources, we decided to conduct an experiment for five days. The experiment was about moral evaluation of an imaginary persuasive system. The experiment was involved six users with various roles including development team, end-user and customer. Furthermore, a set of critical heuristics questions based on Yetim’s (2011b) framework were designed to assess users’ moral reactions from pragmatic, ethical and moral perspectives. During the experiment, participants assumed to play their own role, present answers to pre-defined questions and critically reason with fellow participants. The design and implementation detail of experimentation is carefully documented in Chapter 5.

**Activity 5: Evaluation**

At the end of the experimentation period, a questionnaire was been designed to assess the usefulness and usability of the system. The evaluation of system was examined based on three different data sources including results of the survey, content stored in the system during experimentation period and researcher’s observations. The detailed information concerning evaluation context, assessment method, analysis and findings is documented in this thesis in Chapter 5.

**Activity 6: Communication**

The entire research process and its activities including problem definition and its importance together with designed artifact, its utility, novelty, design and development accompanied by evaluation of its effectiveness, weaknesses and possible future enhancements has been presented in this thesis document.
4. The Software Artifact

For this study, the Rationale Browser as a software artifact was developed to facilitate reflective design and evaluation of persuasive systems. The first section explains how proposed software artifact along with its capabilities might facilitate communication, collaboration, problem comprehension and reasoning during the debate. The second section explores the development process of proposed software artifact including requirement investigation, architecture, design, implementation and artifact demonstration.

4.1 Motivation and Vision

As discussed, moral justification of persuasive systems could be seen as an argumentation process in which different stakeholders reflect their viewpoints and beliefs concerning system’s assumptions, actions, goals and underlying implicit and/or explicit values. The argumentation could originate from system’s consequences, underlying values or assessments of persuasion goals, methods as well as intended or unintended outcome in order to justify its morality (Yetim, 2011b). Similar to the wicked-problems, the problem domain contains both cognitive and social complexities. For illustration, it is difficult to perceive and comprehend problem context because the problem is complex, poorly defined and it has several underlying drawbacks. Also, the problem involves social complexities because different stakeholders with varying goals, objectives and preferences are involved in the problem domain and it is difficult to find an optimal solution to satisfy all needs and viewpoints.

We are proposing an argumentation-based design rationale application to facilitate reflection on ethical aspects of persuasive systems. The ethical conversation implies both communication and rational evaluation of perspectives. In fact, argumentation-based design rationale provides stakeholder with a language of thinking and communication. It brings better structure to the argumentation process and more importantly, it provides stakeholders with a communication vehicle to support rational judgment of resolutions. These aspects created the motivation and vision for development of an argumentation-based design rationale application named as the Rationale Browser. The following section explains the main objectives, concepts and capabilities of the proposed solution.

4.2 The Rationale Browser

The Rationale Browser is an argumentation-based design rationale application developed to support communication and collaboration between stakeholders and facilitate problem comprehension and rational judgment of resolutions. The application is built upon QAR notation model as a language of communication and thinking. QAR is a simplified design rationale method that uses debate and discussion as natural metaphors to represent the argumentation.
QAR notation model expresses argumentation as a collection of questions, answers and arguments. Questions express issues that might have one or more answers as candidate resolutions. Equally, each answer might comprise of one or more arguments that either support or object to it. The Rationale Browser, similar to any other design rationale applications, comprised of two main processes including capturing and representing argumentation. The following diagram (see Fig. 5) explains a holistic view of the solution including its processes, components and their internal interaction.

The application allows stakeholders to effectively pinpoint their viewpoints and reasoning in forms of questions, answers and arguments. The capturing process collects and extracts rationale behind argumentation and stores it to rationale repository while, representation process retrieves information from rationale repository and visualizes it in a way that brings some value for its users.

The QAR notation model could be seen as a language of communication and thinking that enables researchers to effectively capture and represent argumentation in more structured manner. Furthermore, the terminology used in the method is close to human natural language and enables people to easily visualize argumentation in a way that is understandable and easy to use. The application expresses argumentation in hierarchical manner to increase its usability and comprehension. Additionally, collapsible capability enables users to easily switch between summary and detailed view of argumentation.
elements. The summary view shows the essential information only, including title and owner information of nodes whereas, detailed view displays extra information containing nodes description as well as attachments. Altogether these capabilities might strongly encourage sense of collaboration and facilitate mutual understanding and communication among involved stakeholders.

The problem investigation during argumentation process might contain non-linear and complex relationship between issues. In real world situation, one issue might have complex internal relationship with the other. For instance, one question might be closely relevant to other group of questions and likewise, one answer might reference to external resources like an online article. The hyperlink ability enables stakeholders to easily attach associative hyperlinks between elements of argumentation either internal or to external resources. The linking ability also supports stakeholders to easily navigate between relevant questions like traversing a network both forwards and backwards. Moreover, recursive relationship of questions enables stakeholders to make hierarchical relationship between issues. This capability enables them to specialize, generalize and elaborate issues in more efficient way. The linking ability and recursive relationship decline cognitive complexities of argumentation and facilitate investigation and comprehension of complex issues.

The accessibility to the questions is possible through two approaches that include navigation via tree and searching by semantic tags. The application provides users with a navigation panel that showed a list of hyperdocuments along with all relevant questions. The hierarchical view was able to show relevant questions along with their child items in recursive form. This enabled users to easily navigate between various hyperdocuments and questions. Moreover, semantic tagging functionality provided users with an opportunity to attach semantic labels to hyperdocuments or questions. In the later phase, the application uses the semantic tags to effectively explore relevant result based on search criteria defined by users.

Another significant aspect of argumentation-based design rationale applications is to support reasoning behind argumentation. The application enables users to not only present their viewpoints but also express their reasoning behind them. To demonstrate, the users are able to present their viewpoints about a specific topic in shape of answers. The answers contain users viewpoints along with the underlying reasons. In addition, each answer as a candidate resolution might contain one or several arguments that either support or object to it. This could enable critical reflection, reasoning and rational judgment of resolutions.

The Rationale Browser took minimalist approach as a core design principle in both capturing and representation of the argumentation. The minimalistic approach attempts to view the solution as simple as possible without sacrificing essential functionalities. This means that we are eliminating all non-essential features and design noises in order to minimize cost of capturing while maximizing its usability and utility through effective representation. In short, aforementioned capabilities significantly enable communicative and collaborative argumentation and likewise, bring more rationality and clarity to the debate.

### 4.3 The Development Process

This section mainly focuses on the development process of the Rationale Browser. We will start by formulating requirement specification and later explain the software architecture of the artifact. In the next step, we will demonstrate design activities
conducted during early stages of this study. The design activities range from designing user interfaces, modeling behavior with sequence diagrams together with designing the conceptual database schemas for the Rationale Browser. Lastly, we will demonstrate the main functionalities of application and elaborate the source code.

4.3.1 Requirement Specification

The development process started with formal requirement engineering process in order to identify and formulate essential functional and non-functional requirements for the software artifact. The requirement engineering process involves identifying stakeholders as well as their needs, formulating and documenting these requirements for further analysis, design, communication and subsequent implementation (Nuseibeh & Easterbrook, 2000). The requirement investigation phase involved continuous information gathering based on previously reported studies, analyses of the problem domain, discussions as well as meetings with research committee and lastly formulating a requirement specification.

For illustration, the system has various stakeholders including systems administrator and normal users. It requires the system administrator to be able to manage basic information including creation and modification of hyperdocuments as well as semantic tags. Similarly, the users are required to be able to easily register and sign-in to the accounts and subsequently be able to get involved in a conversation by creating a question, posting an answer or argument. Furthermore, the system must provide users with a capability to search questions and browse between various hyperdocuments or questions. And lastly, the application must be capable to visualize argumentation in hierarchical structure to increase its usability and comprehension for users. The following picture shows high-level use case diagram of system (see Fig. 6)
The subsequent table demonstrates high-level list of requirements along with a short description (see Table 1). The rest of this chapter elaborates systems architecture, design and implementation of these requirements.

Table 1. High Level Requirement Specification

<table>
<thead>
<tr>
<th>No</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ-1</td>
<td>Registration and Authentication</td>
<td>The users must be able to easily register and sign-in into their accounts.</td>
</tr>
<tr>
<td>REQ-2</td>
<td>Creating a Question, Answer and Argument</td>
<td>The users must be able to get involved in a conversation by creating a question, posting an answer or argument.</td>
</tr>
<tr>
<td>REQ-3</td>
<td>Mark an answer as final decision</td>
<td>The question owner must be able to mark an answer as final decision.</td>
</tr>
<tr>
<td>REQ-4</td>
<td>Browse between Hyperdocuments/Questions</td>
<td>The users must be able to browse on navigation tree including list of hyperdocuments and relevant questions.</td>
</tr>
<tr>
<td>REQ-5</td>
<td>Search Hyperdocuments/Questions</td>
<td>The users must be able to search hyperdocuments and questions based on semantic hash-tags.</td>
</tr>
<tr>
<td>REQ-6</td>
<td>View Conversation</td>
<td>The users must be able to view conversations in hierarchical structure.</td>
</tr>
<tr>
<td>REQ-7</td>
<td>View Hyperdocuments</td>
<td>The users must be able to view detailed information of hyperdocuments.</td>
</tr>
<tr>
<td>REQ-8</td>
<td>Manage Hyperdocuments</td>
<td>The system administrators must be able to create a new hyperdocument or modify/delete existing ones.</td>
</tr>
<tr>
<td>REQ-9</td>
<td>Manage Semantic Tags</td>
<td>The system administrators must be able to create a new semantic tag or modify/delete existing ones.</td>
</tr>
</tbody>
</table>

4.3.2 System Architecture

Software architecture is an engineering step from abstraction to realization of systems requirements. The process involves systematic study of solution in order to identify its components, their internal interaction and their relationship with external environment (Clements et al. 2010). The main objective is to ensure that designed architecture meets both functional (behavioural) and non-functional (non-behavioural) requirements. The software architecture involves four main activities comprising of architectural analysis, synthesis, evaluation and evolution. Architectural analysis is a process of studying and understanding software requirements whereas; synthesis is a process of architectural construction. In addition, evaluation activities aimed to verify quality of the designed architecture in relation to the original requirements. Finally, evolution activities involve continuous process of maintaining, adapting and improving existing architecture in order to improve its efficiency and reduce its complexity as well. (Hofmeister et al. 2007)

Based on the analysis conducted during early stages of this research, following architecture is proposed to address both functional and non-functional aspects (See Fig. 7). The functional aspect indicates the requirements derived from analysis phase while, non-functional aspects can be used to evaluate the operation of a system including its maintainability, flexibility and ease of development.
The proposed architectural design was built upon three-layered architectural model. At first, the model attempts to distribute complexity of solution over three distinct layers including the web, business and data layers. The workflow starts with a user request to perform a particular task. Subsequently, the system accepts the request and passes it to the lower layers to perform a sequence of activities.

For illustration, the web tier comprises of two well-known design patterns including front controller (dispatcher) and MVC pattern. The dispatcher controller provides a centralized entry point for handling requests (Johnson, 2005). Mainly, it is responsible for taking requests from users and performs common activities like session handling, caching, and input filtering. Subsequently, the dispatcher controller forwards the request to the appropriate web component for further processing.

All web components are constructed based on MVC (Model, View, and Controller) design pattern. The MVC model is an architectural pattern that separates representation of information from business logic. As the name indicates, the MVC pattern comprises of three main elements including a model, a view and a controller. The model represents application data structure and could contain business logic, rules and functions. The view is responsible for preparing the result and representing information in various formats such as a web page or a graph report. However, the controller can be seen as a mediator between users request, model and view. In short, the controller is responsible for (1) taking users requests from dispatcher controller; (2) interact with business tier to perform a sequence of activities like processing a workflow or accessing to the back-end database; (3) update a model and (4) forwarding the model to a particular view object for representing information. (Burbeck, 1992)
The business tier contains a set of business objects, where each of them is responsible to perform a particular business activity. The business activities range from executing a certain workflow, retrieving information or storing new data to the back-end database. In this sense, the business tier might also interact with its lower layer (data tier) to perform database related operations.

The data tier contains two main components including a back-end database management system (DBMS) and an active record component. The back-end database is a relational database management system (RDBMS) that provides a formal structure for storing and managing information in relational tables. However, object-oriented and relational technologies built in different paradigms. They are both mandating a very different worldview. This problem has previously been identified as object-relational impedance mismatch (Ireland et al. 2009). The issue relates to increasing cost of development and maintenance of software components in terms of time and budget. Active Record (AR) is another well-known architectural design pattern that hides the complexity of object-relational impedance mismatch. In fact, AR component automatically converts an object to corresponding relational table and vice versa (Fowler, 2002). This enables the systems to effectively store and retrieve information between object-oriented technologies and relational databases without having to worry about their complexities.

The proposed architecture was built upon continuous changes involving analysis, synthesis and evaluation. The three layered architectural style; separation of concerns and employing design patterns could strongly reduce the complexity and increase the speed of software development as well.

4.3.3 System Design

In simple terms, software design is an iterative process through which requirements are translated into more tangible design artifacts. The process usually involves in depth analysis, conceptualization, modeling and construction of the design artifact. (Pressman & Ince. 1992) The design artifacts range from design specification documents, data modeling, and design visualization to prototype of system under development. This section mainly focuses on the design activities conducted during early stages of this research study. The first part focuses on user interaction design that clarifies how the functionality will be provided through web-interfaces and the second part explains internal interaction of software components thorough description of UML (Unified Modeling Language) sequence diagrams. The last part describes conceptual database design behind the Rationale Browser application.

User Interface Design

During design process of the Rationale Browser application, a couple of mock-ups designed to clarify how the system will provide users with functionalities through the web interfaces. The main aim was to design a minimal user interface and remove all design noises to reduce cognitive load of interaction between users and designed web service. The mock-up interfaces presented in this part is designed through an iterative process involved continuous construction, evaluation and improvement.
Authentication Page

As a fundamental functionality, the users must be able to easily register and sign-in into their account. The authentication page is the first entry point to the system; therefore, we have kept it as simple as possible. The login page only contains a sign-in form that allows users to login into their accounts. There is also a registration link that enables new members to create an account (see Fig. 8).

![Figure 8. An early mock-up of authentication page](image)

Home Page

Once the authentication process is completed, the application redirects user to her home page. The home page of the Rationale Browser comprises of three main panels i.e. a navigation bar, navigation tree and a workspace. The navigation bar is designed as a horizontal toolbar that allow users to navigate between different actions such as home page or search functionalities. Moreover, the navigation tree lists hyperdocuments along with their relevant questions in a hierarchical structure. This enables users to effectively browse between different hyperdocuments and traverse between questions like a tree. Additionally, the workspace panel is the major working area that shows different content depending on which hyperlink was clicked by user (see Fig. 9).
View Conversation

By selecting a question item in the list (presented in navigation tree), the user is able to view the entire conversation in QAR format. As the QAR notation model states, an argumentation comprises of a particular question, one or many answers as well as several arguments that might support or object a specific answer. The following mock-up demonstrates how the system will represent a conversation tree (see Fig. 10).

To increase clarity of argumentation, the designed mock-up shows elements of argumentation in hierarchical structure. Also, variety of colors has been employed to add more semantic meaning to elements of argumentation. For illustration, the question is presented in light blue color whereas; answers are shown in white boxes. Depending on the semantic relationship between an argument and an answer, the argument is colored either green or red. The green color indicates that the argument is supporting parent answer but objecting arguments are represented in red color.

As illustrated in the following mock-up (see Fig. 10), the users are able to create a new answer or an argument by clicking on reply link that is located directly below the message. In addition, users are able to mark an answer as final decision or create a new question as a subclass of original question presented in the conversation.
Adding Question, Answer or Argument

In the previous mockup, we described that the users are able to easily create a new question, provide an answer to a question or make an argument about particular answer. As a matter of fact, pressing reply link will lead user to another page containing a simple form that allows creation of answers or arguments. Moreover, clicking on create a new question link enable users to create hierarchical questions. The node creation form contains several fields that allow users to enter a title, provide a description and attach multiple files to a node (see Fig. 11).
As highlighted in the requirement section, accessibility to the hyperdocuments and questions is possible via two approaches that is browsing by navigation tree or searching them by name or semantic tags. The search panel enables users to effectively find relevant hyperdocuments and questions by using different search criteria. They are able to effectively search hyperdocuments or questions by title or tags. For example, tags are snippets of text that describe characteristics of a particular hyperdocument or question. The following mock-up displays an early prototype of search panel (see Fig. 12).
Aside from the functionalities explained in previous mock-ups, the application contains an admin panel that enables system administrators to effectively manage basic information like hyperdocuments and semantic tags. The administration module has another authentication page and it is only accessible to users who have appropriate permission level. For instance, admin users are able to view list of hyperdocuments and, they are able to add, modify or delete an existing hyperdocument from the system. Similar functionalities will be provided by the system to enable management of semantic tags. The following mock-up exemplifies conceptual design for administration module. The navigation bar contains two major links that enable system administrator to manage hyperdocuments or semantic tags (see Fig. 13).
Figure 13. An early mock-up of administrator panel

Modeling Behavior with Sequence Diagrams

Studying and modeling behavior of a system generally involves analysis and decomposition of use-case scenarios in order to identify systems’ components and their interaction. Sequence diagram is a kind of UML (Unified Modeling Language) diagram that can be used to model the processes, operations, components and their internal interaction within the system. To clarify how internal components interact with each other to complete a particular scenario, a couple of sequence diagrams have been created. However, we are presenting few of the most important scenarios within the system.

View a Conversation

When a user clicks on a question in navigation tree, the system attempts to load the entire conversation and shows it to the user in a hierarchical structure. As shown in the sequence diagram (see Fig. 14), the process involves series of actions. The process starts with a request from user to view a particular conversation. Subsequently, the web component accepts the request and invokes another function on business layer. The business layer use data layer in order to retrieve conversation data from the back-end database. The querying process could include several recursive requests to load all elements of conversation tree. Once the query is completed successfully, the business layer builds a conversation model and forwards it to the web controller. Finally, the web controller shows conversation tree in QAR format.
Adding Questions, Answers or Arguments

As explained earlier, the users are able to create a question for a particular hyperdocument or they might create a sub-question in relation to a specific question. Also, they could get involved in conversation by providing an answer or making an argument. The following sequence diagram explains process of creating a node, which can be in shape of a question, answer or argument (see Fig. 15). The process starts with a web request and subsequently, the web controller invoke another function on business layer. The business layer refines and stores the data in back-end database and then returns the operation result to the web controller. Finally, the web controller updates the conversation and show result to the end-user.
Search Questions by Tags

According to the formal requirements presented in previous section, accessibility to the questions is possible through two approaches i.e. browsing via navigation tree or searching questions based on their tags. For example, each question could have one or more assigned tags that describe its characteristics or scope. Similar to the previous sequence diagrams, the search operation involves interaction of three main components including web, business and data layer (see Fig. 16). The web controller is responsible to collect search criteria from submitted web request and afterwards it invokes search function in business layer. The business layer attempts to make a query to the back-end database and retrieve the search result. Subsequently, the business layer constructs a tree model and passes it to the web controller for presentation.

![Sequence diagram of Search Questions by Tags](image)

**Figure 16.** Search Questions by Tags – Sequence diagram

Data Modeling with Entity-Relationship Diagram

According to Chen (1976) Entity-Relationship Diagram (ERD) is an abstract way of describing a database. The aim is to identify and design entities involved in a particular domain along with attributes and their internal relationships. Data modeling techniques are valuable because they increase mutual understanding among stakeholders and guarantee that designed data model meets functional requirements. The following ERD shows logical schema of the Rationale Browser application (see Fig. 17).
As shown in Figure 17, the designed schema comprises of several entities that shape the backend database for the Rationale Browser application. For illustration, hyperdocument entity acts as a wrapper around questions. Each hyperdocument could have several questions as its child elements and it represents several attributes including a primary key, a name and a short description. Questions, answers and arguments as the basic elements of QAR notation are inherited from node entity. The node entity has several attributes such as an identifier, a title, a description, creation time as well as a reference to its owner user object. Each question might have one or more answers and more importantly, they might have recursive relationship that means that each question might have several sub-questions as well. Furthermore, question entity has a one attribute that keep a reference to particular answer object as final decision.

Answers represent responses to a particular issue. Apart from attributes inherited from node entity, each answer is also contains a reference to its parent question. They could similarly have one or more arguments that either support or object to it. Therefore, an argument entity is comprises of a reference to its parent answer along with a status that explains its relationship to a particular answer.

The questions, answers and arguments could have one or more attachments as supplementary information. In this sense, node entity has one-to-many relationship with attachment entity, which comprises of an identifier together with the name and path to attached resource. Moreover, one-to-many relationship between hyperdocuments and questions to tag entity explains that they can have several tags assigned to them.
4.3.4 Artifact Realization and Demonstration

This section elaborates realization of the Rationale Browser application. The realization phase involved implementation, debugging, evaluation, validation and verification of software components to ensure that it meets expected requirements and design qualities explained in previous sections. We will start with the description of development platform and later, we will demonstrate the essential functionalities of the solution and elaborate the source code behind them.

The Development Platform

The software artifact was built upon a variety of existing software components, best practices and design patterns. The source code was developed based on XAMP platform. For illustration, XAMP is an open-source solution stack consisting of set of integrated software components that facilitate and accelerate development of modern web applications (Dvorski, 2007).

XAMP is a cross-platform solution that is able to run on variety of operating systems. It comprises of three major components including PHP, MySQL and Apache HTTP server. PHP is a popular server-side object-oriented programming language designed for web application development. MySQL is a widely used open-source database management system (DBMS) that enables effective storage and retrieval of data among relational tables. And, Apache HTTP server is a popular web server that provides access to dynamic web resources via HTTP protocol. (Dvorski, 2007)

Moreover, variety of best practices and design patterns (as explained in architecture section) are applied in source-code implementation. The project structure is mainly designed based on three-layered software architecture comprises of several packages described as following:

- **Controllers**: contain all controller classes that are responsible to manage business logic and activities.
- **Models**: contain various active-record classes which represent data-structure of application and they are responsible for storing and retrieving data from relational database management system.
- **Views**: this package holds the view files that are responsible to generate dynamic contents based on user interaction.
- **Components**: containing various utility classes aimed to perform common tasks like session management or working with local disk.
- **Configurations**: consisting of a configuration file that allow system administrator to modify application settings.
- **Static-resources**: this package contains variety of web static resources like images, fonts and style sheet documents.

The open-source solution stack along with appropriate employment of best practices and design patterns could potentially simplify and accelerate the development process of web applications. Furthermore, the project structure explained above will bring more structured source code and likewise, support and encourage separation of concerns during development phase.
Elaboration and Demonstration

In this part, we shall examine main functionalities of the artifact and elaborate the source code for each scenario. Though, giving a detailed account of implementation detail is out of scope of the thesis. In a sense, we are explaining some pieces of the source code that perform the key operations of the Rationale Browser. The complete source-code for application will be made public upon completion of the project.

Scenario #1: User Authentication and Registration

Authentication and Registration are the essential functionality of the Rationale Browser. The users should be able to effortlessly sign-in into their account or register as a new member. The login form is designed as simple as possible and contains one sign-in form that allow users to enter their credential information including a username and password. The registration form is also designed similar to the login page and it enables new members to simply create an account. The following picture shows real implementation of login page (see Fig. 18).

![Login Page](image.png)

Figure 18. Login Page

From a technical perspective, the User entity represents both data structure and behavioral aspects of authentication and registration process. The User entity holds credential information and it also has several functions to control its behavior. For illustration, the rules function aims to validate login information before executing verification process whereas authentication function performs authentication process by sending queries to the back-end database. Additionally, the registration function is responsible to extract users’ credential information from web request and store it as a new user to database. The following code snippet shows the detail of User Entity:
class User extends CActiveRecord{
    public $username;
    public $password;

    public function rules()
    {
        return array(
            array('username, password', 'required'),
            array('username, password', 'length', 'max'=>255),
        );
    }

    public function authenticate(){
        $user = User::model()->find(array(
            'condition' => 'username = :username AND password =:password',
            'params' => array(
                ':username'=>$_this->username,
                ':password'=>$_this->password
            ),
        ));
        if($user != null){
            return true;
        } return false;
    }

    public function register(){
        $_this->username = $_POST['username'];
        $_this->password = $_POST['password'];
        $_this->save();
    }
}

Scenario #2: Home Page (Navigation Tree)

Once the authentication process is successfully performed, the application redirects user to the home page. As demonstrated above, the home page comprises of several panels including a navigation bar, a navigation tree and a workspace. The navigation tree is representing list of hyperdocuments along with their questions in a hierarchical fashion (see Fig. 19). The hierarchical representation of rationale elements along with collapsible feature enables users to effectively traverse among various hyperdocuments and questions. Furthermore, each item in navigation tree represents a link. In a way, clicking on each hyperdocuments shows detailed view of that particular hyperdocument (see Fig. 20). Correspondingly, clicking on each question redirect user to conversation view page.
The hyperdocument view shows detailed information along with a list of relevant questions for a particular hyperdocument. Clicking on new question link that is located directly below the description, allows users to create a new question as a child element for that particular hyperdocument (see Fig. 21).
The Hyperdocument entity is responsible for all above-mentioned functionalities. For illustration, the entity comprises of several attributes including a primary key, a name, and a short description along with list of relevant questions. The view function is responsible to load data from the back-end database and forward it to the corresponding view file to generate the result page. In addition, the second function (GetNavigationTree) is responsible to load all existing hyperdocuments along with their relevant child questions in order to construct the navigation tree data. Finally, the last function (CreateQuestion) takes a new question as parameter and stores it to the back-end database. The following code snippet shows the detail of Hyperdocument entity.

class HyperDocument extends CActiveRecord{
    public $id;
    public $name;
    public $description;
    public $questions = array();

    public function view(){
        $object = HyperDocument::model()->findByPk($this->id);
        $this->render('view_hyper_document',array('object' => $object));
    }

    public function getNavigationTree() {
        $result = HyperDocument::model()->with('questions')->findAll();
        return $result;
    }

    public function createQuestion($question){
        $question->save();
        $this->questions->add($question);
    }
}
**Scenario #3 - Conversation View**

Once the user clicks on a question represented in navigation tree, the system redirects the user to the conversation page. As explained above, the conversation view page represents a list of rationale elements (including questions, answers and arguments) in a hierarchical manner (see Fig. 22). Additionally, a variety of colours have been employed to add more semantic meaning to elements of argumentation. For illustration, supported arguments to a particular answer are decorated with green colour whereas, objecting arguments are represented in red colour. In addition, the final answer as decision for the question is marked in yellow colour.

![Conversation Page – Detailed View](image)

The legend bar that is placed at the top of the conversation explains the underlying meaning of colours relevant to argumentation elements. In addition, the control bar is also comprises two buttons which enable users to switch between summary and detailed view of a conversation (see Fig. 23).
The users are also able to effortlessly create a new question, reply a new answer and make an argument to a particular answer. For illustration, each message box is contains several links that are located at below of it. The reply link enable users to simply create a new answer or argument and likewise, create a new question link allow users to create a new sub-question in relevant to the original question represented in conversation page. Clicking on these links redirect users to a new form that enable users to enter a title, a short description and attach multiple files to a node. In case of creating an argument, the users also need to express the relationship between an argument and other answers presented in conversation (see Fig. 24).

Figure 23. Conversation Page – Summary View

Figure 24. Creating an Argument
From technical perspective, the conversation page comprises of a Question entity along with several Answer and Argument entities. For illustration, Question, Answer and Argument as the basic elements of argumentation are inheriting from Node entity. The Node entity holds the basic information of rationale elements including a primary key, a title, and a short description along with a reference to owner user object as well as created timestamp.

```php
class Node extends CActiveRecord {
    public $id;
    public $title;
    public $description;
    public $owner_user_id;
    public $created_time;
}
```

Question, Answer and Argument entities are all inheriting from Node entity. Therefore, they have basic properties of Node entity and they might also have additional attributes. The following code snippet demonstrates the structure of these entities.

```php
class Question extends Node {
    public $parent_question;
    public $decision_answer_id;
    public $answers = array();
}
class Answer extends Node {
    public $arguments = array();
}
class Argument extends Node {
    public $status;
}
```

As the above code shows, the Question entity has several extra attributes including a reference to parent question object, a reference to an answer as final decision as well as list of relevant answer objects. The Answer entity is also contains collection of relevant argument objects whereas, the Argument entity have a status attribute dedicated to represent the relationship between an argument and its parent answer.

The behavioural aspects of conversation page are implemented in a separate web controller. The Conversation controller has five main functions that are responsible to perform activities relevant to conversation page. These activities are ranging from loading a conversation from the back-end database, creating a sub-question, storing a new answer, making an argument as well as marking an answer as final decision. The following code snippet demonstrates implementation of these activities.

```php
class Conversation extends WebController {
    public $question;

    public function loadConversation($question_id) {
        $conversation = new Conversation();
        $question = Question::model()->findByPk($question_id)
                        ->with('answers')->with('arguments');
        $conversation->question = $question;
        return $conversation;
    }
```
public function createQuestion($subQuestion){
    $subQuestion->save();
    $this->question->addQuestion($subQuestion);
}

public function createAnswer($answer){
    $answer->save();
    $this->question->addAnswer($answer);
}

public function createArgument($answer, $argument){
    $argument->save();
    $answer->addArgument($argument);
}

public function markAnswerAsDecision($answer){
    $this->question->decision_answer_id = $answer->id;
    $this->question->save();
}
}

Scenario #4 – Search Page

The search capabilities are accessible by clicking the search button that is within the navigation bar. The search page comprises of three panels that enable users to effectively search hyperdocuments by name or tags and likewise, retrieve questions based on assigned tags.

![Search Page](image)

Figure 25. Search Page

The functionalities of search page are implemented in the web controller. The Search controller has several functions that are responsible to retrieve information based on search criteria defined by users. These functions attempt to construct a search criteria object based on selected keywords and afterward, they attempt to retrieve relevant result from the back-end database and return them for later representation in the navigation tree. The following code snippet demonstrates implementation of search functionalities:
class Search extends WebController{
    public function searchQuestionsByTag($keywords){
        $criteria = new CDbCriteria();
        $criteria->addInCondition("tags", $keywords);
        $search_result = Question::model()->findAll($criteria);
        return $search_result;
    }

    public function searchByTags($keywords){
        $criteria = new CDbCriteria();
        $criteria->addInCondition("tags", $keywords);
        $search_result = HyperDocument::model()->findAll($criteria);
        return $search_result;
    }

    public function searchByName($keywords){
        $criteria = new CDbCriteria();
        $criteria->addInCondition("name", $keywords);
        $search_result = HyperDocument::model()->findAll($criteria);
        return $search_result;
    }
}
5. Evaluation

The purpose of evaluation activity in design science research is twofold; first to determine designed artifact’s usefulness in relation to the research problem and second aim is to identify weaknesses and areas of improvement for the solution under development (Venable et al. 2012). Both are useful in solving the problem and are fundamental in understanding the problem context (Venable et al. 2012). Therefore, an experiment was planned and conducted to evaluate socio-technical aspects of designed artifact including its overall usefulness and impact on the problem situation. The main objective was to assess the overall usability and utility of designed solution and likewise, identifying open issues and improvements for the next research cycle.

The first section of this chapter focuses on research setting where we shall be discussing the experiment, participants and evaluation method conducted for this study. In the second section, we shall analyze the usefulness of designed artifact and later, we shall explain the results and findings. Finally we shall conclude with a summary section.

5.1 Research Setting

In this section we are elaborating on research setting including experiment objectives, execution, participants and evaluation method applied in this study.

5.1.1 Experiment Objectives

The general objective of this study was to assess the usefulness of an argumentation-based design rationale application to reflective design of persuasive systems. More specifically, we were interested to study how and to what extent the Rationale Browser application might facilitate ethical design and evaluation of persuasive systems. However, study of usefulness of a solution comprises both utility and usability assessment (Grudin, 1992). The utility refers to the effectiveness of systems capabilities, whereas, usability implies the ease of use and learnability of a solution (Grudin, 1992).

As we have investigated in this study, the basic promises of argumentation-based design rationale is to facilitate both communication and reasoning during argumentation process. Therefore, the first objective was to assess the utility of the Rationale Browser in fulfilling its capacities. The utility assessment involved evaluation of both communication and reasoning capabilities. Moreover, we were interested to study usability of a target solution. In this way, the second objective was to measure participants’ perception and overall usability of the Rationale Browser. Both aspects play a significant role in applicability and acceptability of a solution to a particular problem context. Moreover, identifying weaknesses and areas of improvements are very significant in design science research. This information could potentially increase our understanding in relation to the problem context and, could be used as an input for the next research cycle. Therefore, the third objective was to identify open issues and possible improvement opportunities.
5.1.2 Experiment Design and Implementation

In accordance with the above-mentioned objectives, we designed and conducted an experiment. The experiment was about moral evaluation of an imaginary persuasive system. The imaginary persuasive application can be defined as an open innovation system that persuades people to collaborate, share and discusses open issues. The main idea behind the system was to encourage firms to utilize external ideas for the product development, improvement, research and innovation. For illustration, different types of firms like public organizations, small businesses and academic institutions are able to freely register, open up a problem and ask users to present their resolutions and viewpoints. The application employs several persuasive features to encourage firms to use the system and likewise persuading people to cooperate and collaborate with each other. The main objective of system is to increase social innovation and accelerate crowd-sourcing procedures.

The duration of the experiment was five days and it was conducted at OASIS (Oulu Advanced Research on Software and Information Systems) research laboratory in University of Oulu. The experiment was conducted with 6 participants including two IS researchers and four research assistants from OASIS research group. To simulate actual use environment, participants were divided into three different groups including technical development team, businesses (customers) as well as normal users. The development team could be seen as individuals or group of people who are actually involved in the design and development of persuasive application under study. The businesses (customers) are public or private companies who are directly or indirectly involved or affected by the system, whereas, the normal users are people who are using the system to present their viewpoints or resolutions.

At the early stages of experiment, the Rationale Browser application was configured and deployed on a server machine to be accessed by all users. To facilitate the argumentation process concerning ethical evaluation of persuasive system under study, we designed 10 questions based on Critical Heuristics Framework (Yetim, 2011b). Each of these questions was designed with a specific aim to investigate various aspects of ethical conversation on persuasive systems. The following table demonstrates the critical heuristics questions used during the experiment (see Table 2)

During the experiment, participants were supposed to play their own role, present answers to pre-defined questions and critically argument with other members. All communications were done through the system and they were supposed to check their own account at least twice a day. The experiment was conducted successfully and all participants actively engaged in online discussion. Moreover, the daily activity of participants along with bugs and open issues were recorded during the experiment.
Table 2. Practical Discourses and Questions Used During Experiment

<table>
<thead>
<tr>
<th>Question</th>
<th>Type</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 - What are the benefits, harms and outcome value of utilizing the system?</td>
<td>Benefit, Harms and Value Analysis</td>
<td>Day 1</td>
</tr>
<tr>
<td>#2 - What are direct or indirect stakeholders involved in the system? And how the system might affect them? (Positive and Negative viewpoints)</td>
<td>Stakeholder Analysis</td>
<td></td>
</tr>
<tr>
<td>#3 - Does promoting Openness and Social Collaboration increase Innovation?</td>
<td>Pragmatic Discourse</td>
<td>Day 2</td>
</tr>
<tr>
<td>#4 - How the system might bring social benefits? Are there alternative ways to increase collaborative innovation?</td>
<td>Pragmatic Discourse</td>
<td></td>
</tr>
<tr>
<td>#5 - Does promoting Openness and Social Collaboration has a side effect that demotes some other values? What are negative consequences?</td>
<td>Pragmatic Discourse</td>
<td>Day 3</td>
</tr>
<tr>
<td>#6 - How the system might bring economical benefits? And how the system might enable unfair competition and abuse of rights?</td>
<td>Pragmatic Discourse</td>
<td></td>
</tr>
<tr>
<td>#7 - How active collaboration and open innovation might increase quality of product/services? And how the system might violate privacy/security/rights?</td>
<td>Pragmatic Discourse, Ethical Discourse</td>
<td>Day 4</td>
</tr>
<tr>
<td>#8 - How collaborative innovation as a value of system defined? Are there other values that have conflict with this value?</td>
<td>Ethical Discourse</td>
<td></td>
</tr>
<tr>
<td>#9 - Are the values promoted are in accord with norms? What are the conflicts?</td>
<td>Moral Discourse</td>
<td>Day 5</td>
</tr>
<tr>
<td>#10 - Is the system good for all? What are the conflict, risks and negative consequences?</td>
<td>Moral Discourse</td>
<td></td>
</tr>
</tbody>
</table>

5.1.3 Evaluation Method

At the end of experimentation, a survey was designed and has been completed by all participants. The questionnaire comprised of 13 Likert scale questions and one open question to collect users' feedback and suggestion for further improvement. The questionnaire was made based on the experiment objectives and each question aimed to investigate particular aspect of solution including its utility and usability. For illustration, the questions relevant to the utility assessment were focused on verifying basic capabilities of proposed solution. These questions aimed to verify how the solution facilitated communication, reasoning and problem comprehension among stakeholders. The second section of questionnaire emphasized on usability aspects of the system such as simplicity and ease of use. Finally, the last question was designed as an open-ended question to collect users' feedback and suggestions.

To determine effectiveness of solution in relevance to the above-mentioned aspects, we are studying them by using Goal Questions Metric (GQM) method (Caldiera et al., 1994). The GQM is a measurement mechanism for evaluating software features. For illustration, the goal statement is explaining the ultimate objective of a particular software feature whereas, question and metric aims to characterize the assessment and achievement of a goal (Caldiera et al., 1994).
5.2 Evaluation of the Rationale Browser

In this section, we are examining the systems effectiveness in fulfilling its promises. The solution have been designed with aim of making communication more transparent, argumentative and explicit to its stakeholders and likewise, supporting rationale judgmental of resolutions. Therefore, communication and reasoning are the basic promises of the Rationale Browser and more importantly, they are playing very significant role in reflective design process. However, simplicity and ease of use of the solution are also important. In this sense, we are examining usefulness of the Rationale Browser from two angles including utility and usability perspectives. Furthermore, in the last part we will also elaborate on artifact weaknesses and open issues based on observation and analysis of users' feedbacks.

5.2.1 Utility Assessment

The utility assessment indicate to the examining both communication and reasoning capabilities. For illustration, communication specifies system's capabilities in providing a communication bridge between stakeholders with conflicting perspectives whereas; reasoning refers to the capabilities to support rationale judgment of viewpoints.

*Elaboration* - The application could be seen as a communication vehicle between stakeholders, which enables them to effectively express their concerns and viewpoints in form of questions, answers and arguments. The QAR notation model behaves as a shared language of communication and thinking among stakeholders and likewise, it enable system to effectively capture and represent elements of argumentation in more structured manner. This might strongly increase mutual understanding and accelerate communication among involved stakeholders. Furthermore, the hyperlink ability enables stakeholders to attach associative hyperlinks and likewise, parental relationship between questions enables them to simply specialize, generalize and elaborate issues in more effective way. These features might decline cognitive complexities of argumentation and facilitate investigation and comprehension of complex issues. Moreover, the solution also enables stakeholder to effectively express their reasons behind viewpoints. Each answer as a candidate resolution might contain one or several arguments that either support or object to it. This might strongly enable critical reflection, reasoning and rational judgment of resolutions.

*Analysis* - To clarify effectiveness of proposed solution in relevance to the above-mentioned capabilities, we examined perceived effectiveness of such aspects from participants' point of view. The following section shows GQM table (see Table 3) relevant to communication and reasoning capabilities.
Table 3. GQM – Communication and Reasoning

<table>
<thead>
<tr>
<th>Goal</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>- To facilitate communication among various stakeholder with different perspectives</td>
<td></td>
</tr>
<tr>
<td>- To facilitate problem comprehension and investigation</td>
<td></td>
</tr>
<tr>
<td>- To facilitate representation of reasons behind resolutions</td>
<td></td>
</tr>
<tr>
<td>- To facilitate rationale judgment of resolutions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Does system enable stakeholders to present their viewpoints and opinions?</td>
</tr>
<tr>
<td></td>
<td>- Does system facilitated capturing of discussion between stakeholders with different roles?</td>
</tr>
<tr>
<td></td>
<td>- Does system facilitated navigation and investigation between questions?</td>
</tr>
<tr>
<td></td>
<td>- Does system enable stakeholders to present their reasoning behind resolutions?</td>
</tr>
<tr>
<td></td>
<td>- Does system facilitated rational judgment of resolutions?</td>
</tr>
<tr>
<td>Metric</td>
<td>- Measured by perceived effectiveness from participants' point of view</td>
</tr>
</tbody>
</table>

The summary of statistics from survey responses shows that the solution was decently successful in fulfilling its promises regarding communication and reasoning capabilities. Most of the participants reported that the solution facilitated capturing of discussion among stakeholders with different perspectives. The solution enabled them to effectively present their viewpoints and concerns regarding to pre-defined questions. In addition, most of participants were agreed that hierarchical representation of argumentation helped them to effectively navigate between issues, answers and arguments. Moreover, over half of participants were reported positive feedback regarding reasoning capabilities. The solution allowed them to effectively present their reasons behind their viewpoints and enabled them to rationally assess other resolutions.

The following figures (see Fig. 26 - 28) show the perceived effectiveness of system in relevance to the communication and reasoning capabilities.

Figure 26. Perceived Effectiveness – Communication Capability
5.2.2 Usability Assessment

The usability assessment focuses on the overall system's simplicity and ease of use. There are two major aspects that are very important in this domain including system notation model used during debate as well as hierarchical representation of argumentation.

Elaboration – The solution was built upon QAR notation model. The QAR is a simplified design rationale method that uses debate or discussion as a natural metaphor to represent the argumentation. The terminology used in the method is close to human natural language in order to be understandable and easy to use. The QAR notation model is employed as language of communication both for capturing and representation of argumentation. In this sense, the first objective of usability assessment is to evaluate the ease of use and simplicity of QAR notation model. Furthermore, the application represents elements of argumentation in a hierarchical structure. The conversation page shows selected question along with list of answers and arguments in hierarchical manner. The collapsible capability enables users to easily switch between summary and detailed view of argumentation. The conversation page is one of the major components of solution. Therefore, the second objective of usability assessment is to evaluate the ease of use and usefulness of hierarchical representation of arguments.

Analysis - To clarify usability of proposed solution in relevance to the above-mentioned aspects, we examined perceived usability of those aspects from participants' point of view. The following section shows GQM table (see Table 4) relevant to usability aspects of the Rationale Browser.
Table 4. GQM – Usability

<table>
<thead>
<tr>
<th>Goal</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- System must provide users with an easy to use and simple notation model for capturing of argumentation.</td>
<td>- System must provide users with an easy to use and simple approach for representation of argumentation.</td>
</tr>
<tr>
<td>Question</td>
<td>Question</td>
</tr>
<tr>
<td>- How was the overall usability of QAR notation model?</td>
<td>- How was the overall usability of hierarchical representation of argumentation?</td>
</tr>
<tr>
<td>Metric</td>
<td>Metric</td>
</tr>
<tr>
<td>- Measured by perceived usability from participants' point of view</td>
<td></td>
</tr>
</tbody>
</table>

The summary of statistics from survey responses shows that participants rated the overall usability of system high. Most of the participants reported that system notations (question, answer and argument structure) were simple and easy to use. In addition, almost all of participants in this study agreed that hierarchical view of argumentation was both useful and easy to use. The following pictures (see Fig. 29 and 30) show the overall perceived usability of system.

**Figure 29. Perceived Usability – Hierarchical Visualization**

**Figure 30. Perceived Usability – QAR Notation Model**
5.2.3 Open Issues

We carefully inspected data from daily activity of participants and users' feedbacks in order to identify open issues and weaknesses of the Rationale Browser. Correspondingly, we have identified six major issues that can be classified into functional and usability problems. Here we elaborate more on them in the following subsections.

Usability Issues

Issue #1: There should be more clear distinction between stakeholders and their roles in the system.

The system currently does not provide any mechanism for the users to distinguish and understand stakeholders' roles involved in a conversation. The users must be able to specify their role during registration procedure and likewise, the system must be able to show this information in conversation page. This minor improvement might strongly increase clarity and sense of collaboration among users.

Issue #2: There should be a more distinctive way to separate supporting and objecting arguments than the color of the title.

Depends on semantic relationship between an argument and answer, the title of argument is decorated with green or red colour. However, there should be a more distinctive way to separate supporting and objecting arguments than the colour of the title. The system is not able to separate arguments when the title is blank and also it is really difficult for the colour blind to see the difference. Therefore, rather than changing colour of the title, it is more effective to decorate the message box with a background colour.

Issue #3: The question you are answering should be shown when adding an answer. This is also valid for adding new argument; the user must be able to see what he/she is answering to.

The basic information about parent object must be shown in node creation form so that users can see and understand what he/she is answering to.

Issue #4: Navigation in long conversation is difficult and confusing

Navigation in long conversation is both difficult and confusing. The system must provide users with a mechanism to be able to quickly jump into various section of conversation. From technical perspective, this is feasible to implement by using HTML internal anchors that allow users to jump into various section of a particular web page.

Functional Issues

Issue #5: Adding validation to node creation form (Question, Answer, Argument form) is essential. Title must be a mandatory field.

New validation constraint is required for node creation form. The title must be a mandatory field and system required to populate it with some default values.
Issue #6: The navigation panel (tree view) is not working well on all browsers.

This is due to the incompatibility issues among web-browsers. The tree view component required more consideration and it should be compatible at least with the latest version of popular web-browsers.

5.3 Summary

The main aim of evaluation activity in design science research is to determine designed artifact usefulness in relation to the research problem and likewise, identifying weaknesses and areas of improvement for the solution under study. The result from experiment presented in this chapter, indicated that the solution was quite successful in fulfilling its promises. However, there are many possibilities for further improvement. Therefore, there is a need to refer back to the design and development phase in order to improve usefulness of the artifact. To increase reliability and validity of next research cycle, the future experiment can be carried out in an actual use environment with more participants and with real persuasive application under evaluation.
6. Discussion

We are now living in a fast-pace technological world. Technology has significantly affected our lives, businesses, industry and the way people communicate and interact with each other. Information technology is never neutral (Oinas-Kukkonen & Harjumaa, 2009). They are all trying to influence our attitude and/or behavior for several motives including promoting socially valued activities like healthier lifestyle, promoting safety in society, environmental conservation and many more. These were just a few of many examples of persuasive technologies. Yet, promoting moral goals via technologies raises many ethical concerns (Spahn, 2012).

As we have explored, moral justification of persuasive systems can be seen as an argumentation process in which stakeholders reflect their viewpoints and beliefs concerning system’s assumption, action, goal and underlying implicit or explicit values (Yetim, 2011b). The argumentation process involves both exchanging of viewpoints as well as assessment of persuasion goals, methods, outcome and values in order to justify its morality. Therefore, similar to the wicked-problems, the argumentation process contains both cognitive as well as social complexities. It means that the problem is difficult to perceive and comprehend and more importantly, it involves various stakeholders with conflicting goals, preferences and interests.

In regard to the main research question, the ultimate goal of this study was to assess the usefulness and usability of an argumentation-based design rationale to reflective design of persuasive system. More precisely, we were interested to study how and in what extent the Rationale Browser application might facilitate ethical design and evaluation of persuasive systems.

The major challenges relevant to the reflective design of persuasive systems can be classified into communication, reasoning and problem comprehension issues. These aspects are playing very important roles in ethical conversation and likewise, they are main promises of argumentation-based design rationale. The Rationale Browser application was designed with 3 aims in mind; (1) to make communication more transparent, argumentative and explicit to its stakeholders; (2) to support rationale assessment of viewpoints and (3) to decrease cognitive complexity of problem domain by structuring argumentation and making its comprehension easy for stakeholders.

The proposed solution can be seen as a discourse-support tool that accelerate active argumentation concerning identifying and checking actions, goals and values from pragmatic (purposive), ethical (good) and moral (just) perspective. The perceived usefulness and usability from participants’ point of view confirm that the solution decreased cognitive and social complexities of argumentation and likewise, increased quality of both communication and reasoning among stakeholders. The solution establishes a rationale and democratic communication bridge among the parties and enabled them to effectively argument and assesses motives behind target persuasive application under evaluation. Obviously, these aspects do not guarantee moral persuasive systems, but these are essential prerequisites that are needed for moral justification process of persuasive systems. However, from practical point of view, there are many other concerns that required more considerations.
Consideration 1 - Conflict with Business Interests

From business side point of view, discursive perspective to ethical design and evaluation of persuasive systems required more resources both in terms of cost and time. Obviously, involving users during design process of persuasive system is very valuable for designers. However, since there is no tangible software artifact, the users of persuasive systems do not have deep understanding about the application and its actions, goals and values. Therefore, moral evaluation of persuasive system during use-time is essential but this is very costly process since changing action, goal and value of persuasive system required heavy technical modification.

Furthermore, in a real world scenario, one aspect of system might have complex internal relationship with other aspects. This means that there might be a co-relationship or dependency between actions, goals and values of persuasive system under evaluation. Therefore, any modification to underlying actions, goals and value of persuasive system might leads to cascading impact on design of system under evaluation. This may causes several costly changes for the application provider.

Consideration 2 - Challenges with Shared Communication Language

As we have discussed earlier, ethical design and evaluation of persuasive systems involves various stakeholders with different worldview and conflicting objectives. Therefore, one can claim that supporting unified language of communication and thinking is essential component of any discourse-support applications. The Rationale Browser application was built upon QAR notation model. The minimal notation model employed in our solution effectively facilitated capturing and representation of argumentation.

In addition, another important aspect is how to guide argumentation among conflicting perspectives. This is exactly where Critical Heuristic Framework proposed by Yetim (2011b) comes into play. The practical discourses along with designed heuristic questions promote transparent and rational argumentation for ethical conversation either during design process and use time of persuasive systems. In this study, we acknowledge that apart from above-mentioned issues, creating shared vocabulary among stakeholders is very important. The designed questions during argumentation must not use any jargon or difficult vocabularies. The questions must be designed as simple as possible and each question must explicitly focuses on one and only one specific aspect of persuasive system under evaluation.

Consideration 3 – Applicability of Discourse Ethics

Habermas, in his seminal book called Between Facts and Norms (1996), stated that the discourse ethic theory is applicable in any type of consensus oriented group especially in democracy planning and legislative processes. In fact, discourse ethic was developed upon communicative rationality theory, which states that moral justification is only possible by involving all affected stakeholders in a rationale and open argumentation process. However, Løgstrup et al. (1997) rejected the possibility of evaluating ethics based on action and consequences and emphasized that ethics must be considered intuitive. In the other word, he defined ethic as an intuitive object rather than moral rule based on reasons.
Furthermore, in contrast to the universalism perspective, relativism argues that moral norms and values are strongly depends on several variables and it is not possible to consider them universally. From moral perspective, this means that the outcome of discourse process is applicable in only limited context. Moreover, persuasive systems may leads to variety of consequences which may affect direct and indirect stakeholders in both positive and negative manner. In this sense, it is not possible to easily claim that “all” affected stakeholders including indirect stakeholders were involved in moral justification process. And the last but not least, it is never possible to fulfill condition of an “ideal speech situation” in discourse processes. Therefore, from realistic perspective, rationale argumentation concerning moral evaluation of persuasive systems may leads to only partial morality which is applicable in a limited context.

Conclusive remarks

With above-mentioned, ethical conversation is the most effective and practical solution concerning moral evaluation of persuasive systems. However, it does not mean that other moral methodological approaches like guideline-based and stakeholder analysis methods are useless. The moral evaluation of persuasive systems must be integrated in entire process from early stages of design to use-time of target persuasive system. In this context, several moral methodological frameworks might be employed during the entire design process and use time of persuasive systems. In this sense, the VSD framework as a comprehensive and iterative process model can be served as a unified umbrella for all activities.

More importantly, ethical conversation per se is not leading to moral persuasive system. The ethical conversation only provides a method to find optimal norms that regulate among involved stakeholders. The proposed solution in this study is aimed to facilitate reflective design and evaluation of persuasive systems by providing a rationale and democratic communication bridge among the parties.

Suggestion for Future Study

We have already identified and presented six major open issues for the Rationale Browser application. Therefore, the first step for the next research cycle would be fixing those mentioned minor issues and afterward, conducting another experiment in an actual use environment with more participants and with real persuasive application under evaluation. Here we are proposing some future enhancements that might strongly increase overall usefulness of argumentation-based design rationale applications. Some of them aim to increase quality of both communication and reasoning during argumentation process and others focuses on usability aspects of system.

Suggestion 1 – Semantic Analysis and Sentiment Evaluation of Semi-Structured Argumentation

The main challenge of DR approaches can be describe as an optimization problem, which aims to minimize the cost of capturing and meanwhile maximizing its usability via effective representation (Regli et al., 2000; Shum & Hammond, 1994). In contrast to informal models, formal schemas are more structured and strict in capturing and representation of rationale whereas semiformal models combine advantages of both approaches. In this sense, the main question is how we can actually improve effectiveness of semiformal methods without increasing overhead on capturing of rationale elements.
We propose to extend semantic meaning of rationale elements by using hash-tags. The hash-tags are snippets of text that describe characteristics of a particular entity. This technique can be used in annotating argumentation including unstructured data especially for attachments and multimedia resources. This might help us to effectively classify complex issues like a network of topics connected with set of keywords. In the later phase, the users will be able to effectively search and navigate among groups of relevant topics. The application might employ two different approaches for annotating argumentation. The first approach allows users to manually attach set of pre-defined keywords to argumentation whereas the second approach enables automating annotating of argumentation by using semantic analysis techniques. Semantic analysis can be defined as application of text mining and natural language processing (NLP) techniques in order to identify syntactic structures of argumentation, its elements, relationship and it’s meaning as whole.

For illustration, semantic concept extraction technique enables us to simply extract more frequently used keywords during argumentation. These keywords can be seen as main aspects of a specific conversation and they might describe the whole meaning of argumentation. The application might use these keywords in order to automate annotating of argumentation and in later phase, it will enable users to effectively search topics based on their semantic content.

Apart from above-mentioned capabilities, sentiment analysis (opinion mining) technique might enable us to automatically analyze messages and determine the attitude of its owner in relevance to a specific topic. This means that the application will be able to automatically extract semantic attitude of messages in forms of positive/negative/neutral viewpoints. This information can be used to classify the relationship between an answer and its arguments in QAR notation model.

**Suggestion 2 - Social Rating System**

Social rating systems are kind of collaborative system that allows users to collectively evaluate quality of a particular entity. In other words, this feature enables users to simply give positive or negative points for each answer. This might strongly encourage sense of collaboration among stakeholders and likewise improve reasoning capability of argumentation-based design rationale applications. For instance, in our solution, the final answer can be marked as a decision. However, in many cases, similar problems might arise but due to the many reasons the final marked solution as decision might not applicable to similar cases. In this sense, it would be effective to list all relevant answers to a specific question and sort them by their positive rates.

**Suggestion 3 - Persuasive System Design**

Improving usefulness of software by adding more features usually comes with cost of usability. This means that adding new functionalities to software might make it more useful but it might also increase cognitive load of interaction and make solution difficult to use. In the same way, making an application easier to use usually required removing some functionality that might make it less useful for its users. However, both of these aspects including usefulness and usability are important in knowledge creation applications. (Räisänen & Oinas-Kukkonen, 2007)
In this sense, persuasive system design (Oinas-Kukkonen & Harjumaa, 2009) can be considered as a solution. For illustration, argumentation-based design rationale applications might utilize various persuasive design principles in order to motivate users to contribute actively in the argumentation. For instance, persuasive techniques like rewarding might actually motivate users to contribute more by adding new answer or argument. This might actually make argumentation-based design rational applications more collaborative and more fruitful for its users.
7. Conclusion

The motivation of this study started from the fact that information technology is never neutral and they are all trying to influence our attitude and/or behavior in one way or another. In this thesis, we have examined process of construction and evaluation of an argumentation-based design rationale application with aim of facilitating reflective design and evaluation of persuasive systems.

As theoretical background, we have studied moral challenges of persuasive systems and afterward we discussed several methodological frameworks and viewpoints concerning moral evaluation of persuasive systems. As we have explored, moral justification of persuasive systems is possible only by involving all concerned parties in a rationale and open argumentation process. In fact, ethical conversation on persuasive systems enables stakeholders to effectively argument and assesses motives behind the system under evaluation and likewise, it allows designers to understand user’s moral reaction to the designed technology. Nevertheless, involving users during moral justification process might bring many challenges along with its benefits. Similar to the wicked-problems, reflective design of persuasive system contains both social as well as cognitive complexities. The argumentation process involves various stakeholders with conflicting perspectives and more importantly, the problem domain is difficult to comprehend and resolve. In the second section, we have presented the design rationale theory along with its approaches. We elaborated on argumentation-based design rationale and how it can actually improve quality of argumentation process.

Concerning research method, due to the nature of this research, we have employed DSR methodology. This study followed the formal sequential procedure initiated from problem identification and motivation. The problem has been recognized by prior studies and the main objective of solution was to facilitate reflective design and moral evaluation of persuasive systems. The adopted research methodology along with its activities was comprehensively explained in this document.

To tackle with aforementioned challenges, we have developed an argumentation-based design rationale application. The Rationale Browser was designed with three goals in mind; (1) to make communication more transparent, argumentative and explicit to its stakeholders; (2) to support rationale assessment of viewpoints and (3) to decrease cognitive complexity of problem domain by structuring argumentation and making its comprehension easy for stakeholders. The solution establishes a rationale and democratic communication bridge among the parties and enables them to effectively assess the system from pragmatic (purposive), ethical (good) and moral (just) perspectives. The design and development of artifact was followed by formal software engineering process comprising of requirement engineering, high-level architecture, design, and implementation and verification process. The solution was designed upon three-layered software architecture model and several best practices including MVC and ActiveRecord were adopted during implementation phase. The design activities and detail of implementation including source-code of solution was comprehensively elaborated in this document.
To evaluate usefulness of designed artifact in relation to the research problem, we have designed and conducted an experiment. The main objective of experiment was to evaluate the usefulness and usability of the solution and likewise, identifying open issues and possible improvements for the next research cycle. The perceived utility and usability from participants’ point of view confirm that the solution decreased cognitive and social complexities of argumentation and likewise, increased quality of both communication and reasoning among involved stakeholders. Moreover, based on our observation and analysis of users' feedbacks, we have also presented six major open issues related to the Rationale Browser.

We started our discussion by presenting three major practical considerations including conflicts of discursive approach with business interests, challenges in making shared communication language and practically of ethical reflection in moral justification of persuasive systems. We also acknowledged that moral evaluation of persuasive systems must be integrated in entire development process from early stages of design to actual use-time of product. In this sense, the VSD framework as a comprehensive and iterative process model can be served as a unified umbrella for all activities.

The major implication of this study is to reveal that argumentation-based design rationale can strongly decrease complexity of argumentation process and likewise, improve quality of both communication and reasoning among involved stakeholders. These aspects per se do not guarantee moral persuasive systems, but these are essential prerequisites that are needed for moral justification of persuasive systems. From practical perspective, we have developed and evaluated a discourse-support application that effectively establishes a rationale and democratic communication bridge among the parties. The application is not tied to any specific problem domain and it can strongly mitigate complexity of any consensus-making processes. Theoretically, we acknowledged that three major aspects of argumentation-based design rationale applications including communication, reasoning and problem comprehension capabilities have a significant effect on mitigating social and cognitive complexities of reflective design and evaluation of persuasive systems. Finally, based on our observation, we have presented some future enhancements that might actually make argumentation-based design rational applications more collaborative and more fruitful for its users.

This study is also constrained by several limitations. The small number of participants as well as using imaginary persuasive application for the experiment was the main limitation of our work. Therefore, future studies are required to confirm our findings. To increase reliability and validity of next research cycle, the future experiment will be carried out in an actual use environment with more participants and with real persuasive application under evaluation. Future work will also aim at improving overall usefulness of designed artifact by fixing open issues and implementing proposed enhancements. We hope that our work will be a step forward in development of moral persuasive applications.
References


Batya Friedman. (1997). Human values and the design of computer technology *Cambridge University Press*.


Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity, 41*(1), 37-54.


Whelton, M., Ballard, G., & Tommelein, I. (2001). Application of design rationale systems to project definition–Establishing a research project. 9th Conference of the International Group for Lean Construction (IGLC), Singapore, August, , 10

