

Adding Ethical Values to Situated Cognitive Engineering

ARVIND MOHABIR, 1324810

Master Media & Knowledge Engineering

Specialization: Man Machine Interaction

Faculty EEMCS, TU Delft

January 16, 2013

Preface

Thesis committee

The thesis committee consists of Prof. dr. Mark Neerincx, dr. Tjerk de Greef and dr. Ir Ibo van de Poel. Mark Neerincx is a full professor at the Interactive Intelligence group of Computer Science at TU Delft and a senior researcher at TNO. Mark is one of the founding fathers of the situated Cognitive Engineering (sCE) methodology and one of the frontrunners on the related sCET tool. Tjerk de Greef is a post-doc researcher at the Intelligent Intelligence group of the faculty of Electronic Engineering, Mathematics and Computer Science at TU Delft and currently works on the project 'military human enhancement: Design for Responsibility and Combat Systems'. Ibo van de Poel is an associate professor in Values and Technology at the philosophy department/group of the faculty of Technology and Policy Management at TU Delft and teaches a Value Sensitive Design course at his faculty.

Acknowledgements

I would like to thank Tjerk de Greef for his (almost) daily input and assistance and Mark Neerincx for any other input and assistance, when Tjerk was not available. Thanks to Christian Detweiler, I could find tons of articles on Value Sensitive Design and other related subjects on ethical values. He also suggested to have contact with Ibo van de Poel. I would like to thank Ibo van de Poel for his input and willingness to take place in my committee.

I would like to thank the other sCET frontrunners Jurriaan van Diggelen, Olivier Blanson Henkemans and Joris Janssen at TNO for helping me out with designing and testing sCET and letting me participate in the sCET meeting for the roadmap of the tool. My regards to the sCET developers Jens Eldering and Ruud van den Beukel for providing me with the code and helping me out with implementation and installation problems.

Finally I would like to thank Mark Hendriks, Alex Dirkszwager, Elmer Jacobs, Sebastiaan Meijer, Allart Hoekstra, Rien Kostanje, Steven Bos, the guys from Hogeschool Rotterdam and many more which I probably have forgotten to mention for their company at the masters graduation room on the twelfth floor.

ARVIND MOHABIR

JANUARY 16, 2013

Summary

Systems are getting more intelligent and emergent potentially impacting ethical values. Such ethical values need to be accounted for early in design process. VSD is a design practice that specifically focuses on ethical values. However VSD lacks an explicit and systematically transfer towards a design rational, which is addressed explicitly in the situated Cognitive Engineering methodology. The topic of this master thesis is to combine the best of both worlds by combining VSD with sCE towards a new design methodology called sCEthics. As such, the research question reads:

HOW COULD SITUATED COGNITIVE ENGINEERING AND VALUE SENSITIVE DESIGN BE COMBINED TO ADDRESS VALUES IN A SYSTEMATIC MANNER?

Both VSD and sCE have their roots in user-centered design (UCD), but have a different focus. VSD focuses on the elicitation of values and stakeholders and possible trade-offs between values which is supported by qualitative and quantitative evaluations. sCE focuses on the specification of a system with requirements elicitation at its core which are justified by claims and contextualized by use cases. The requirements baseline is improved by iterating through evaluation with stakeholders. sCE's advantage relates to a better definition of the methodology compared to VSD, while the latter methodology has a strong emphasize on values.

Both a literature review and interviews show that sCE and VSD complement each other nicely. The combination of VSD and sCE is called sCEthics. However, six requirements are implemented and evaluated. 1) All investigations of VSD are integrated in the foundation of sCE in sCEthics. To help bridge the gap between the value elicitation of VSD and the requirement elicitation of sCE, both 2) policies and 3) an extended scenario system have been added to the new methodology. 4) Design patterns have been introduced to find solutions on how to incorporate requirements into the design of a system. The last big inclusion involves the visualization of the data in sCE to create a better overview of the data. Two forms of visualization are implemented: 5) a matrix overview and 6) a radial visualization. The matrix overview shows the users to which requirements and ethical values a claim, a use case or a scenario is linked. The radial visualization gives a more general insight into the connections within a project.

sCEthics was evaluated using eleven participants. Each participant needed to execute six tasks, which were related to six new requirements that were embedded in sCEthics. In addition, the participants answered questionnaires on the usefulness and positive and negative claims of the new requirements. The results reveal that the implementation of requirements relating to values, policies and design patterns are seen as useful, but needs revision to reduce the workload for the sCEthics user. The extended scenario system was not seen as very useful, due to the benefits of the scenario system not outweighing the amount of extra work necessary. Both visualization additions were seen as useful because the overviews created a better insight to the connections within a project and also showed which connections were still missing. Neither of the two visualizations gave the participant any large strain on their information load, but both did give them a nice overview. The radial visualization was

seen as more useful than the matrix overview, because the radial visualization gave the participants a better overview of the whole project at first sight.

Concluding, sCETHics can be seen as a good step towards the creation of a methodology combining requirements analysis with value elicitation. As can be concluded from the results, five of the six requirements were seen as useful but some small improvements to the implementation are mandatory. The requirements elicitation of sCE and values elicitation of VSD complement each other. By addressing values and possible trade-offs earlier on, the requirements baseline can be extended to address these oversights. sCETHics makes it easier to trace back requirements to the values important to stakeholders.

Contents

1	Introduction	8
2	Background	11
2.1	Values.....	12
2.1.1	Relation between law and value.....	13
2.1.2	Should a specific value just have a single definition?.....	14
2.1.3	Elicitation and evaluation of values.....	14
2.1.4	Values in robot ethics.....	15
2.2	Design Methodologies	17
2.2.1	Value Sensitive Design.....	17
2.2.2	Cognitive Engineering.....	19
2.2.3	Situated Cognitive Engineering	20
2.2.4	Comparison of methodologies.....	21
2.3	Conclusion	22
3	Design of sCEThics	25
3.1	Case: ePartners that care	25
3.2	Foundation	25
3.2.1	Operational demands.....	25
3.2.2	Human factors.....	26
3.2.3	Envisioned technology.....	26
3.2.4	Additional requirements.....	26
3.3	Specification stage	27
3.3.1	Use cases and scenarios	28
3.3.2	Claims	29
3.3.3	Requirements.....	30
3.3.4	Core functions.....	30
3.3.5	Additional requirements.....	30
3.4	Evaluation stage.....	30
3.5	Requirements	31
3.6	Conclusion	33
4	Implementation.....	35

4.1	SCET History ~ Different versions of sCET	35
4.1.1	Earliest version of sCET ~ sCET prototype	35
4.1.2	Fundament for a new version of sCET ~ sCET 1.0	36
4.1.3	Current version of sCET ~ sCET 2.0.....	36
4.1.4	Advantages & disadvantages	37
4.2	The core of sCEThics ~ leading to new opportunities	37
4.3	Modular system of sCET	39
4.4	Adding overview to sCET	41
4.4.1	Matrix overview	41
4.4.2	Radial Visualization.....	42
4.5	Discussion	43
5	Evaluation.....	45
5.1	Goal of evaluation.....	45
5.2	Literature on testing usefulness	45
5.3	Type of evaluation	46
5.4	Procedure for evaluation.....	47
5.5	Participants.....	49
6	Results.....	50
6.1	Pre-task questionnaire	50
6.2	Task questionnaires	50
6.2.1	Grading usefulness	50
6.2.2	Measuring claims	53
6.2.3	Possible improvements to the implementation	54
6.3	Post-task questionnaire.....	55
6.4	Possible future work	56
7	Discussion.....	58
7.1	Values.....	58
7.2	Policies.....	59
7.3	Scenario system	59
7.4	Design Patterns.....	59
7.5	Matrix overview	60
7.6	Radial Visualization	60

7.7	Possible future work	60
7.8	General observations	60
8	Conclusion	62
8.1	Research question.....	62
8.2	Future work	63
8.3	Take away message.....	63
9	References.....	64
10	Table of figures	66
11	Table of tables	67
12	Appendix 1: Expert reviews.....	68
12.1	Expert Interview: Ibo van de Poel	68
12.2	Expert interview: Jurriaan van Diggelen.....	69
12.3	Meeting Military Human Enhancement.....	70
12.4	Follow-up meeting Jurriaan van Diggelen	72
13	Appendix 2: Tables and figures of the evaluation	73
13.1	Experience Questionnaire	73
13.2	Task Questionnaires	74
13.2.1	Values	74
13.2.2	Policies	74
13.2.3	Scenario system	74
13.2.4	Design patterns	75
13.2.5	Matrix overview	75
13.2.6	Radial visualization	75
13.3	Post-task questionnaire.....	76
13.4	Possible future work	76

1 Introduction

Values can be described in multiple ways. (Friedman, Kahn, & Borning, 2003) One definition is based on the monetary definition in which it is described how much one would pay for an object. Another definition is based on importance of an object, describing how useful or important an object is to someone. This is related to the principles of a person to act *rightful* and do *good*. The latter type of value is used throughout this thesis and relates to importance and principles. This type of value has to be taken into account in the design of autonomous systems.

Autonomous and intelligent systems are being used more and more in risky domains such as the military and health domain. When designing systems for such domains, it is important to safeguard ethical values that are important in these sensitive domains by the application of technology. Not addressing ethical values in the design and specification, could potentially lead to an undesirable system. Examples of ethical values are human welfare, privacy and safety when military drones are deployed or values like trust, ownership and human welfare in electronic health records for hospitals and practitioners. Examples are the use of military drones in Afghanistan and Pakistan by the US army and the design of electronic health record in the Netherlands.

In a development plan for the period 2009 – 2047, the US air force states that the next generation of drones will have a higher level of autonomy (The Economist, 2011), implying that the drones could take action without interference from users. The US air force uses drones in Afghanistan and Pakistan for surveillance and targeted killings. Drones are airplanes without a physical pilot inside. Current systems use pilots at a remote location to fly these drones. In addition to a number of technical advantages, there are also several ethical values affected by the use of drones.

Potentially impacted ethical values include responsibility and accountability. Increasing the autonomy of the drone would decrease the human factor of the decision and legal and moral issues would surface. For example, a drone with a high level of autonomy could make a decision to bomb a building, which might contain terrorists. Who is responsible for this decision and how is this decision related to military law? In what way can it be moral for a computerized system to decide to attack on its own? Does the military have any guidelines for computerized systems? These values will have to come to mind, while designing such an unmanned system.

The medical domain serves as another domain where ethical values play an important role. The implementation of electronic health records is a prime example of wrongfully addressing ethical values in design and implementation. The system allows digital access to health records of patients. This would allow for a quick exchange of information between different health instances and insurance companies. Conflicting values between stakeholders including the health department, hospitals, practitioners and insurance companies were at the base of the failed implementation of the system.¹ For example the practitioners and the insurance companies had a conflict about the ownership and privacy of the records.

¹ *Electronisch Patientendossier gaat definitief niet door (Dutch)*, <http://www.volkskrant.nl/> Last visited on: 26-11-2012

Values like ownership, trust, privacy and safety played a huge role in the issues behind the huge delays in the electronic health records project. Questions were related to the security and access of the system, possible abuse by insurance companies, the correctness of the information and the ownership of the records. Due to these problems, the plan for implementing the electronic health records was rejected by the Dutch senate.² Proper and systematic identification of these issues could address these concerns earlier on to improve the quality of the system and create a higher acceptance for the system among stakeholders.

Design methodologies are a systematic approach to develop new software products. User centered design (UCD) is a design philosophy which focuses on creating a new system in cooperation with a user.³ Value Sensitive Design (VSD) and Situated Cognitive Engineering (sCE) are design methodologies that are based on the UCD philosophy.

Value sensitive design is a design methodology which incorporates values in the design of a system. (Friedman et al., 2003) The method is based on a tripartite methodology which consists of three parts: Conceptual, empirical and technical investigations. In short, VSD focuses on the elicitation of ethical values and stakeholders through empirical research with value conflict resolution among stakeholders.

There are no strict rules in which order to use the VSD methodology, which can be either an advantage or a disadvantage for the designer. There is a list of practical suggestions how to use this methodology. One of the strengths of this methodology would be the identification of direct and indirect stakeholders, values and conflicts. Another advantage is the focus on empirical investigation to get a better insight of the stakeholders with qualitative research and a better overall impression with quantitative research.

sCE takes the activities and needs of the user and context into account (Neerincx & Lindenberg, 2008; Neerincx, 2010). The methodology contains three main parts consisting of the foundation, specification and evaluation. It is an iterative type of methodology in which the specified requirements are constantly being refined based on the results of the tests. The strength of sCE is seen in the systematic specification of requirements with their design rationale, the strong focus on contextual information and the constant iteration through empirical research to improve the requirements and design to the likings of the users.

Both design methodologies also have their weaknesses. VSD leaves the specification and design of the system up to the designer and the loose nature of the methodology can make it hard for a user to decide where to start and what to do. The empirical investigation could also lead to a longer design phase. sCE is more strict in its way of working and does not explicitly take ethical values into account in the design of a new system along with no differentiation between direct and indirect stakeholders. It is also not possible (yet) to explicitly document quantitative and qualitative research in the foundation like the evaluation stage does with the human-in-the-loop and prototype testing.

² *Eerste Kamer legt elektronisch patientendossier stil (Dutch)*, <http://www.nrc.nl/> Last visited on 26-11-2012

³ *What is User-Centered Design?*, <http://www.usabilityprofessionals.org/>, Last visited on 23-12-2012

The two methodologies would complement each other nicely with their strengths, while eliminating some weaknesses. The looseness of VSD is compensated in the sCE by clear definition of the three parts, while the ethical values and empirical investigation of VSD can be integrated in sCE. It should be possible to combine these methodologies to incorporate the value elicitation of VSD with the requirement elicitation of sCE to create a new methodology using the strengths of both. The structure of sCE should be extendable with the investigations of VSD, especially in the foundation stage of sCE.

This leads to the following main research question:

HOW COULD SITUATED COGNITIVE ENGINEERING AND VALUE SENSITIVE DESIGN BE COMBINED TO ADDRESS VALUES
IN A SYSTEMATIC MANNER?

An online tool called sCET has been designed at TNO to use sCE on projects. It is possible to document the different phases of sCE inside the tool and expansion of the system is possible allowing to extend sCET with ethical values as used in VSD.

This first step at combining sCE and VSD should be seen as a stepping-stone towards the both ethical and systematic design of an autonomous system. The ethical and systematic design of this new methodology should help design a better system with a higher satisfaction level to the involved stakeholders.

To answer this research question, several steps need to be taken. First up, we need to identify what ethical values are and explain both methodologies in more detail. Next up, requirements with both positive and negative claims will need to be specified for the new combined methodology and a new design will have to be made. Subsequently, these requirements need to be implemented in sCET. Afterwards the system is evaluated with expert users of the tool to examine the usefulness of the new methodology together with the positive and negative claims of the requirements. Finally the results are accumulated and the research question should be answered.

The following research points will be addressed in the upcoming chapters to answer the main research question. First, a literature research will be done to examine values and design methodologies in chapter 2. This will be followed by the design of the new methodology in chapter 3. Next up will be the documentation of the implementation of the changes into the sCET tool (chapter 4). Subsequently in chapter 5, the evaluation procedure will be explained. Thereafter the results of the evaluation will be shown and interpreted (chapter 6). In the following chapter 7, the discussion, the results will be discussed based on the expectation and comments. Finally, there will be a conclusion in chapter 8 containing an answer to the main research question, possible future work study and small take away message.

2 Background

The US military uses unmanned air vehicles (UAV) in Afghanistan mainly for surveillance and bombing missions. UAVs are smaller compared to traditional aircraft fighters because the pilot is not required to be the fighter and operates the plane from the ground instead (The Economist, 2011). Having no personnel on board implies that drones are lighter, are more fuel efficient and can stay longer in the air. The aircrafts are flown at a remote location by a trained officer from a computer screen with a joystick.

In the future it is expected that advanced technologies allow these drones to fly (partially) autonomously and cooperate with a remote pilot as a joint cognitive system. The reasoning behind using remote piloting and increased levels of autonomy relates to minimizing the human casualty rate by outsourcing dangerous missions (Strawser, 2010) while increasing the safety risks of the pilots. By shifting the staff off the plane and to a location off base would cause the planes to be smaller, which implies a much lower cost. The expected deployment of drones with a high level of autonomy means that a drone would have to inhibit a level of artificial intelligence. The UAV's then have the ability to think and act independently, which potentially requires the UAV to decide on moral issues, such as the decision to kill a civilian affecting on the integrity of a human being targeted and poses a responsibility question regarding the decision. These ethical values have to be taken into account while designing such high levels of autonomy.

As the name implies, User Centered Design (UCD) is a design philosophy with a large focus on the users. A new system is designed and validated in cooperation with the users. Both Value Sensitive Design and Situated Cognitive Engineering are forms of user centered design implying that the new system should agree to the needs of the users implying a better satisfaction among the users. A disadvantage of this methodology is the amount of time necessary to design and test the system.

Value sensitive design is a design methodology that preferably starts by defining ethical values at stake by using the system. In short, VSD subscribes to find the core values and stakeholders, find empirical evidence to support the values and find the right technology given the core values. VSD has a strong emphasis with ethical values, but the methodology does not have a strong emphasis on actual specification of a new system.

Situated cognitive engineering is an expansion on cognitive engineering. Cognitive engineering bridges the gap between the user and the researcher by constantly iterating and testing with the user, until it converges into a satisfactory result. This could also be seen as a form of user centered design, but it has a different point of view by focusing on requirements elicitation rather than on ethical values and stakeholders. Situated cognitive engineering has additional elements concerning the context.

This research focuses on combining situated cognitive engineering with value sensitive design. This leads to a design methodology that A) embeds ethical values, B) shows how ethical values translate to requirements, and C) displays how ethical values relate to the implementation and testing the system. The goal of this chapter is to establish background knowledge on affected ethical values in autonomous systems.

As such, this chapter addresses the following questions:

- How can values be defined?
- How can values be used to design autonomous systems?
- What ways are there to resolve conflicts between values?
- What are the requirements for incorporating values into SCE?

First, values will be explored in the first section. Then the connection between ethics and autonomous systems will be examined. Subsequently, design methodologies to create agents will be explored with an emphasis on values and users. Finally the conclusion will contain answers to the questions stated above.

2.1 Values

Before going into more detail on what we understand a value to be, definitions given by dictionaries or encyclopedias are discussed.

Cambridge dictionary

Value noun (IMPORTANCE)

[S or U] **the importance or worth of something for someone**

For them, the house's main value lay in its quiet country location.

*They are known to **place/put/set** a high value **on** good presentation.*

[U] **how useful or important something is**

*The photos are **of** immense historical value.*

*His contribution was of little or no **practical** value.*

*The necklace had great **sentimental** value.*

*It has **novelty** value because I've never done anything like it before.*

Oxford dictionary

1 [mass noun] the regard that something is held to deserve; the importance, worth, or usefulness of something:

2 (values) principles or standards of behaviour; one's judgement of what is important in life:***they internalize their parents' rules and values***

Merriam-Webster

1: a fair return or equivalent in goods, services, or money for something exchanged

2: relative worth, utility, or importance <a good ~ at the price> <the ~ of base stealing in baseball> <had nothing of ~ to say>

3: something (as a principle or quality) intrinsically valuable or desirable <sought material ~**s** instead of human ~**s** — W. H. Jones>

All three dictionaries make a distinction between multiple definitions of the word value. The first definition of the word would be related to the cost of a product or a service, a monetary value to say so. The first description of Merriam-Webster conveys that definition of value. The other two definitions of the word are related to the relative worth of something and are shown in the second and third description of Merriam-Webster. This something can either be an object or a principle.

The relative worth of an object is referred to as the extrinsic or **instrumental value**, such as the sentimental value of a necklace, as shown in the second definition from the Cambridge dictionary. This definition of value is associated with the subjective value related to physical or abstract objects.

A value based on principles is called the **intrinsic value** (e.g. trust and privacy) and is the type of value this research will focus on. This type of value is easily identifiable, but it is hard to find justifications for why it is good or bad. The second description from Oxford and the third description from Merriam-Webster convey the meaning of intrinsic values rather nicely. Intrinsic value does not apply on objects, but are principles that stand on their own. These values are based on principles, which are found over time by society or religion. Other names used for intrinsic value are moral value, values with ethical import (Friedman & Kahn, 2003) and value claims (Schroeder, 2008). Throughout this study we will use intrinsic values.

The relationship between intrinsic and instrumental values is rather simple. From each instrumental value, an intrinsic value can be derived by further reasoning why this instrumental value is important. It is possible to derive instrumental values from intrinsic values. This would imply that there is a (mutual) connection between values. Research has also been done on the account of ordering values in the way of a value hierarchy (Poel, 2012). sCE benefits from at least connecting values and perhaps ordering values. **Therefore SCE should contain a way to construct connections between values.**

2.1.1 Relation between law and value

Intrinsic values are about what is right and wrong. Laws are the do's and don'ts given to us by the government or state. (Dix, 2008) Laws and intrinsic values are not the same, but they are related. Values are personal to someone, while the governmental institutes form laws through values important within a society. Something being legal does not necessarily mean that it is moral. An example of this would be bullfights in Spain. This disgusts many people, but it is legal. When there is a conflict of interests among stakeholders, the justice system is used to find a sufficient solution to which all stakeholders can agree upon. The same would hold for conflicting values. This justice system consists of addressing the conflict from all stakeholder sides and one or more judges coming to a conclusion. While this approach is not optimal, it seems to be a good solution to address incommensurability (having no common standard) until a general theory about value conflict resolution exists (Nagel, 1979).

Figure 1 shows a value hierarchy created by Ibo van de Poel (Poel, 2012). The values hierarchy shows that norms would help make design requirements from values. Norms are not the same as values, but the two are related. While values describe what ideals and motives we as a society aim for in pursuit of happiness, norms explain how we should behave in certain contexts to collectively achieve these ideals. These norms are not necessarily written down, but can over time be standardized in policies by

governmental or corporate instances in terms of laws and restrictions. While not all policy documents are written down in clear text, enough policies can be concrete enough to be used as requirements in the new design. **Therefore the new methodology should take norms and policies into account.**

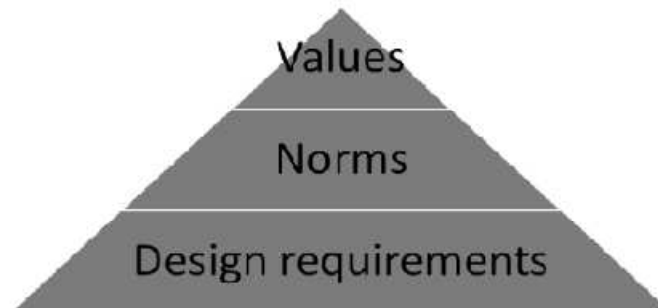


Figure 1: Values hierarchy by Ibo van de Poel (Poel, 2012)

2.1.2 Should a specific value just have a single definition?

Can values have multiple definitions? This is a question raised by many philosophers. The group answering positively is called pluralist while the other group is referred to as monist (Schroeder, 2008). Monists say pluralists are either '*explanatorily inadequate*' or have not found the basic understanding yet. Pluralist say monists are too strict about their definitions and this could deny addressing a value formally, while it belongs to a value intuitively. An important issue between the two groups is based on incommensurability, an inequality measure for theories. This term is closely related to incomparability. From a pluralistic point of view, theories are incommensurable, because there is no common ground with multiple definitions for a single value. Monists say that practical wisdom requires for difficult choices to be made, despite the level of complication.

For within a project, the sCE methodology should use a monist stance to value definition to have and maybe even force a consensus among all stakeholders. A pluralist point of view would be desirable over projects, so different definitions of an intrinsic value can be handled between different projects. This means that a single value can have different definitions for a value for separate projects, but not within a single project.

2.1.3 Elicitation and evaluation of values

Ways for the elicitation of values are disputed. There are small breakthroughs in this process. Photo elicitation is used more often to extract values from the user (Dantec, Poole, & Wyche, 2009; Pommeranz, Detweiler, Wiggers, & Jonker, 2011). In short, photo elicitation uses photos instead of words to tell a story. Based on these pictures, users will try to tell, for example, the emotions these pictures evoke. These pictures can be filed by the future user group. This is mostly dependent on the context and usage of a new product and the prior knowledge of the user. The idea of this method is that pictures say more than words.

The reason of using methods like photo elicitation marks a shift to understand values which users deem important in specific contexts. Other studies (Borning & Muller, 2011; Cheng & Fleischmann, 2010) also seem to imply this shift. This requires a more in-depth method of inquiry. Quantitative methods like

surveys do not lend themselves for these types of inquiry, meaning the usage of more qualitative methods like interviews. The major disadvantage of qualitative methods is the amount of time and resources necessary. An advantage would be a better understanding of the values in play. **Therefore the new methodology should include qualitative methods to identify values in the foundation and include qualitative evaluation methods to evaluate these values with the stakeholders.** To start, interviews and group discussions should be included. Photo elicitation can be added at a later date.

2.1.4 Values in robot ethics

Agent and robot ethics describe the in what way agents and robots can be (made) ethical and is relevant to discuss regarding the ethical design of autonomous systems. Intelligent autonomous systems are called agents and intelligent robots are physical embodied agents. Robot ethics is not an entirely new concept. Famous science-fiction writer Isaac Asimov has stated three laws of robotics to use in his novels back in the 1940s in his *Robot* series of short novels including *I, Robot* (Murphy & Woods, 2009). His three laws were:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

While these laws were used strictly as a story-telling device by Asimov, robot ethicists have taken inspiration from these laws. Asimov's laws assume that the robots in his fictional universe are at an intelligence level that is similar to humans. Scientists and researchers are not even close to that at this moment of time. Murphy and Woods (Murphy & Woods, 2009) describe an alternative set of laws for robots to abide to. This alternative set contains laws that are situated and could be achieved right now. This does not mean that the set does not take the laws of robotics from Asimov into account, but adjusts it to current times. Their three laws are:

1. A human may not deploy a robot without the human-robot work system meeting the highest legal and professional standards of safety and ethics.
2. A robot must respond to humans as appropriate for their roles.
3. A robot must be endowed with sufficient situated autonomy to protect its own existence as long as such protection provides smooth transfer of control to other agents consistent the first and second laws.

Their rules take into account the current state of robotics by situating robots to a specific environment and they put a large emphasis on safety and responsibility from the designers. The Institute of Electrical and Electronics Engineers (IEEE), the Association for Computing Machinery (ACM) in America and the British Computer Society (BCS) are taken in high regard for the regulation. At this moment, there is no code of ethics or code of conduct available for specifically the development of agents or autonomous robots. There is a code of ethics for software engineering (Gotterbarn, Miller, & Rogerson, 1999) and

decision making (Anderson, Johnson, Gotterbarn, & Perrolle, 1993), which should contain at least a part of robot ethics. The code of ethics for decision making is used in the 'killer robot' case (Epstein, 1994) to illustrate the mistakes made by the company designing the titular robot. Bear in mind that these codes are all but complete. The field is constantly changing, which makes it hard to predict which codes are going to be relevant in the future. The codes are generic, which makes them more of a guideline than a strict rule set. These codes can be directly linked to the values hierarchy including the norms. The codes are loose policies to which a system should abhor and therefore can be used to specify requirements.

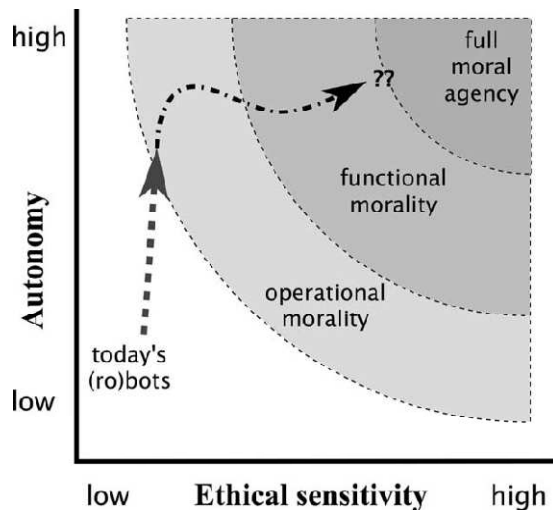


Figure 2: Two dimensions of Artificial Moral Agent development

Several robot ethicists (Moor, 2006; Wallach & Allen, 2009) make a distinction in the type of morality and these types are similar in design. Wallace and Allen call agents with regards to some form of ethics, artificial moral agents. As shown in Figure 2, Wallach and Allen (Wallach & Allen, 2009) base their model on two dimensions being autonomy (how solitary a robot is allowed of capable to operate) and ethical sensitivity (to what level does the robot take ethical issues into account, for example decision support systems like a medical ethics expert system). These two levels are independent.

At the lowest level is the operational morality, which implies that ethics are taken into account in the design of the agent. This would be servable for agents with a low level of autonomy with a high level

of ethical sensitivity or a low level of ethical sensitivity with a high level of autonomy. The next level of morality is the functional morality for agents with at least a moderate level of autonomy and sensitivity. The morality is not only taken into account in the design, but also in development the agent. This implies that the agent would be able to decide for itself, on the morality of an action. The highest level of morality is full moral agency of which humans and humanoids are an example.

Moor's model (Moor, 2006) is in essence the same as the model by Wallach and Allen, but it makes an additional distinction. Below the four levels of Moor's model are listed.

1. Ethical-impact agents
2. Implicit ethical agents
3. Explicit ethical agents
4. Full ethical agents

Ethical-impact agents are agents that impact the ethics in the used environment. In the article by Moor (Moor, 2006), the example of camel riders in the middle east has been taken. The boys riding these camels are treated like slaves. By replacing them with robots, boys are most likely saved from enslavement, thus having an ethical impact on the environment. On the other hand, Implicit ethical

agents have an implicit set of ethics and are similar to the agents with operational morality. Ethical considerations are taken into account in the design of the agent but the agent does not have any ethical reasoning capabilities. An example would be to take sustainability and safety into account, while designing a new car. The car is ethical by design, but it is not smart enough to act and reason ethically. Thirdly, explicit ethical agents are similar to agents with functional morality. The agent then has the ability to act and reason ethically. The ethical aspect is implemented explicitly. Full ethical agents are basically humans and humanoids.

Both papers agree that the field should not be focusing on full ethical agents and should instead focus on explicit ethical agents/functional moral agents. In the end, the development of explicit ethical robots could lead to full ethical robots.

An important note to make is the distinction between the use of human ethics and robot ethics. What holds for a human being, does not have to hold for an agent or robot. For example in the case of military robots, staying alive does not have the same impact for a robot as it has for a human soldier. (Arkin, 2008)

Inducing ethical values into systems is called artificial morality by Wallach and Allen (Wallach & Allen, 2009). Artificial morality is seen as a way to study human ethics in the same way as artificial intelligence is a study of human intelligence. These two fields are not mutually exclusive and according to Wallach and Allen, should be done concurrently to further advance our understanding of humans.

2.2 Design Methodologies

In this section, different design methodologies will be described. First, value sensitive design (VSD) will be explained and after that cognitive engineering (CE) is explained. This is followed by explaining situated cognitive engineering (sCE). Finally there will be a comparison of the methodologies, weigh both the pros and the cons of VSD and sCE.

2.2.1 Value Sensitive Design

Value sensitive design, VSD in short, is a design methodology to develop systems, while keeping values important to (in-)direct stakeholders into account. However, VSD has a specific focuses on moral values. (Friedman et al., 2003) The methodology is a tripartite methodology:

- ***Conceptual investigation***

This investigation consists of the identification of ethical values and the direct and indirect stakeholders, followed by finding trade-offs between conflicting values. Values can be derived either through lists of intrinsic values or through instrumental values from stakeholders. Direct stakeholders are actual users of the new system, while indirect stakeholders do not use the system, but are related with it. To avoid conflicts between the interests and values of the stakeholders, trade-offs will have to be made between the different parties.

- ***Empirical investigation***

The empirical investigation is about the observation, measurement and documentation of human activity. To facilitate these tasks, quantitative (for statistical purposes) and qualitative (for in-depth information) methods can be used. Examples of these methods are respectively

questionnaires and interviews. Other more practical examples are rapid prototyping and human in the loop experimentation. It is a useful way of testing the theory with practice and can be used as a validation or further study of the conceptual investigation.

- **Technical investigation**

The technical investigation checks the availability and effectiveness of current or near-future technology. The available technologies are then tested against the values found in the previous investigations. The most high-tech technology available does not have to be the best suited technology given the values and stakeholders. For example, high-quality cameras with limited privacy and safety options could have a worse effect on the privacy of the user than a camera of slightly lesser quality video with more options. A proactive design will have to be made with the values in mind.

Friedman has made an (incomplete) list of universal values. These values are Human Welfare, Ownership and Property, Privacy, Freedom from Bias, Universal Usability, Trust, Autonomy, Informed Consent, Accountability, Identity, Calmness, and Environmental Sustainability, Courtesy. A definition of these values (including examples) can be found in the paper by Friedman (Friedman & Kahn, 2003).

There is no strict guideline to use the methodology in a specific way. Friedman (Friedman et al., 2003) has a few practical suggestions. It is suggested to follow the order of the conceptual investigation, the empirical investigation and the technical investigation.

Several aspects of VSD are disputed (Borning & Muller, 2011; Dantec et al., 2009). These disputed aspects include the use of universal values, the order of execution and the loose nature of the methodology.

The use of universal values and the order of execution are closely related due to context. By first defining these values in the conceptual investigation would increase the bias of direction for the empirical investigation. A solution would be to start with an empirical investigation to find the values of the users and continue with the conceptual investigation using the found values. Advantages of this process are the value elicitation from the stakeholders and the reduction of bias for the researcher on values. A disadvantage would be the (lack of) definition of the values by the stakeholders. This is defined to be part of the conceptual investigation and would mean that there should be a small investigation beforehand or that both investigations should be done concurrently.

Due to the loose nature of the methodology, there is not much direction or guidance in using VSD. This could especially harm the empirical investigation, where there are a lot of options to do both qualitative and quantitative research. Suggestions have been made to address more direction in which methods to use within the empirical investigation. As such, Borning and Muller (Borning & Muller, 2011) suggest a method of photo elicitation in the empirical investigation for a less technical informed crowd. Dantec et al. (Dantec et al., 2009) are thinking on a higher level and suggest that research methods should be specified to both the research and crowd. Values will need qualitative methods for elicitation for a better understanding of core ethical values. It is suggested that research should make their voices and/or bias clear in both the research and the empirical trials. The actual investigation would be similar

to the co-design element of participatory design, integrating the user into the development process. Table 1 contains a summary of the advantages and disadvantages of Value Sensitive Design.

Advantages	Disadvantages
Takes values in to mind	Less emphasis on the actual design than on values
Non-linear / Iterative	Universal values might be too restricting and are not context based
	Not very strict means less documentation

Table 1: Advantages and disadvantages of VSD

2.2.2 Cognitive Engineering

Cognitive engineering is described by Norman (Norman, 1986) as a type of applied cognitive science, attempting to apply scientific knowledge to the design and implementation of new physical systems. It tries to find the conceptual model of a certain task. This concerns the goals a user has to achieve to do a certain task and the actions a user makes to achieve these goals. Figure 3 illustrates the execution and evaluation bridges of CE. To execute an action in the execution bridge, the system has to know what the intentions are concerning the goal, which actions are possible and finally has to do the action with the help of an interface mechanism. To evaluate an action, the system has to receive information from the environment through an interface, interpret this information and finally evaluate whether this action was beneficial in achieving the goals.

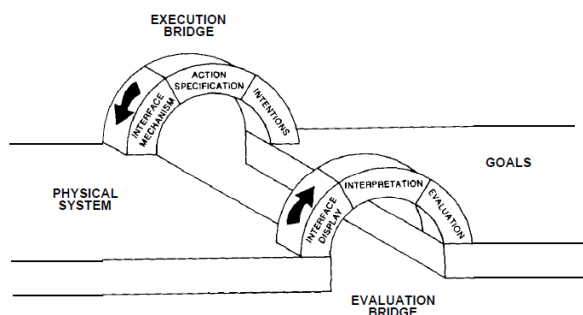


Figure 3: Bridging the gulfs of execution and evaluation

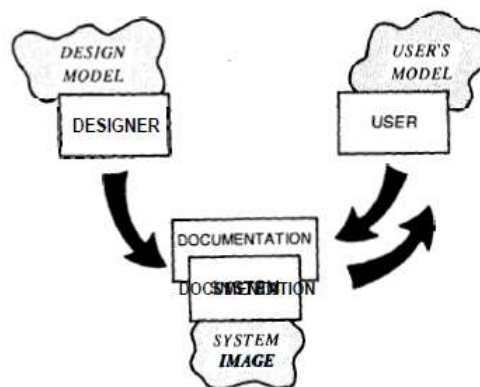


Figure 4: The three conceptual models

There are three different conceptual models to be considered in Figure 4. The first one would be the design model and is the model that the designer believes the system should do. This is the conceptual model held by the designer. The next model is the user's model formed by the user when/by using the machine. The final model called the system image is formed from the actual implementation and documentation of the system. The user's model should be as close as possible to the design model. The system image should make this happen by interaction between the system image and the user's model.

The limitations of a physical system might not make it (fully) possible to implement all changes to agree to a theoretically perfect design model. Trade-offs will have to be made to justify design decisions based on technology, time, effort and expense. The trade-offs will have to be discussed with the stakeholders

to find a common ground. Norman states that approximate methods should suffice for most applications. This is in a way similar to the level of abstraction by Floridi and Sanders (Floridi & Sanders, 2004), which states that by confining the context clears up vague ideas and definitions. Floridi and Sanders also state that the levels of abstraction abstract the environment not overlooking its vital importance, but merely acknowledging its lack of immediate relevance to the current discourse. Simply put, minor details might be important, but may not be of high relevance to a new system right now. Approximations agree with this statement in that they leave out minor details, while focusing on more important aspects.

2.2.3 Situated Cognitive Engineering

Situated cognitive engineering (Neerincx, 2010) or sCE in short is a methodology created at the TNO research institute during research on the Mission Execution Crew Assistant (MECA) project for future manned deep space missions. The methodology was created due to a lack of concise methods to develop envisioned human-automation systems. sCE combines user-centered design with cognitive engineering and requirements analyses. The outline of the methodology is shown below in Figure 5.

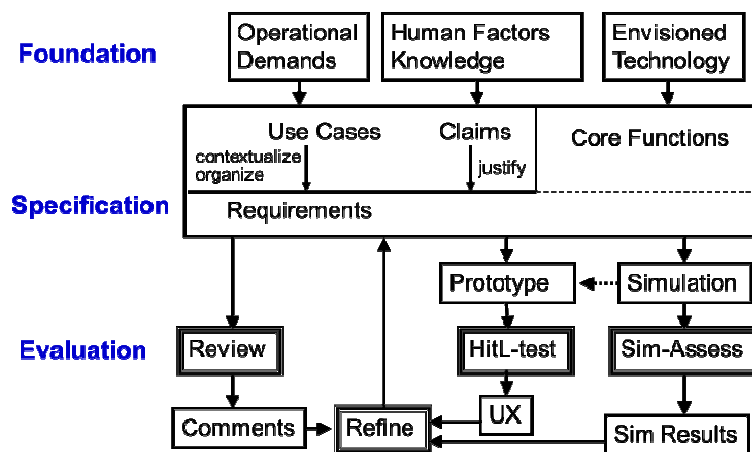


Figure 5: Situated Cognitive Engineering

The foundation describes the situational information, including the operational demands, human factors and envisioned technology. Operational demands examine external factors including context, roles, tasks and processes in the field. Human factors describe the internal factors including theories and determinants. The envisioned technology finds out what technology is available currently or in the near future and which positive and negative arguments users have towards a certain technology.

The specification focuses on building a requirements baseline through use of use cases claims and core functions. Core functions describe the high level functionality of a system defined by the designers and stakeholders. Use cases are used to describe the context of use for one or more requirements. Claims justify a requirement by stating the positive and negative effects of a requirement. Requirements describe what the new system should contain.

In the evaluation the requirements are tested and evaluated with users using the claims. Testing is done through simulation and prototyping. The review is done in cooperation with the stakeholders. Based on

the tests and evaluation, the requirements baseline is refined. This loop is closely related to the refinement of the conceptual model of CE. The requirements baseline could be seen as part of the system image in cognitive engineering. The baseline is iteratively checked and modified, which makes for a useful analogy to the system image.

Table 2 contains the advantages and disadvantages of sCE. Requirements elicitation in cooperation with the users is central to sCE. The methodology has a strict definition in phasing the development in three different stages, all stages of design have been covered and development is done iteratively. It should be noted that users do not have to start at the foundation. They can create their own workflow within sCE. The downside of the methodology is lack of use of ethical values and the lack of actual documentation of the methodology. Most papers on sCE are written from an application point of view rather the theoretical base of the model and why the methodology works theoretically rather than empirically.

TNO has developed sCET to support the employees with the development of their product using situated cognitive engineering.

Advantages	Disadvantages
User is central in the design	No (strict) use of values
Start to finish methodology	Not very well documented (Where are the papers about sCE, not the examples of sCE-designed software?)
Iterative	

Table 2: Advantages and disadvantages of Situated Cognitive Engineering

2.2.4 Comparison of methodologies

Both VSD and sCE can be described as user-centered design methodologies, but the difference is mostly in the direction. VSD is directed towards values while sCE focuses on eliciting requirements.

VSD seems to define a few must-haves, but leaves the rest to the researchers in filling in the how. sCE seems to be stricter in usage. Especially the derive and specify stage are clearly defined to follow.

A relevant question relates to the strictness of the methodologies. While VSD is very loose in actual usage of the methodology, sCE is strict in that is a start to finish methodology with clearly specified foundation, specification and evaluation stages. Common practice among sCE users seems to be an emphasis on detailing essential sections (requirements, use cases and claims) and shortly describing the obvious sections. Adding flexibility could help improve the usage of sCE. This could be done by having a selection of templates or allowing the users to add the relevant sections themselves with a few mandatory parts. A requirement then should be to allow flexibility for the user within the blocks of sCE.

The structure of sCE might also be its strength, because the framework allows for easier expansion than for example extending VSD. sCE also already has some functionality that seems to have at least some resemblance to the investigations in VSD. Incorporating the rest of the functionality of VSD in sCE should not require a complete overhaul of sCE. That is why we will continue with sCE as the core methodology for expansion.

2.3 Conclusion

In this conclusion, the questions raised in the introduction will be answered. There might be an overlap in the answers of the questions.

HOW CAN VALUES BE DEFINED?

A distinction is made between two types of values. The first type of value is called instrumental value and suggests the relative worth of an object. The other type of value is the intrinsic value and its definition is more vague. Intrinsic values are values that do not have something to tell about an object, but focuses more on principles. It is also said that intrinsic values can be regarded as ends-in-itself. This means that the values are not important because of something else, but it is important because of the satisfaction of itself. For example, privacy is an intrinsic value, because someone finds it important on its own and not because of something else.

There is no strict consensus on the amount of definitions of a single value. Philosophers are split up in two camps. The first camp is monist and concludes that a value can only have a single definition. The other camp is pluralist and says that a value can have multiple definitions.

HOW CAN VALUES BE USED TO DESIGN AUTONOMOUS SYSTEMS?

While designing agents, values can be taken into account in several ways. Both Moor (Moor, 2006) and Wallach and Allen (Wallach & Allen, 2009) describe levels of ethics for agents. The descriptions are rather similar with the only large distinction being the first level of Moor's model, called the ethical impact agent. At this level, the usage of an agent impacts its environment. The next level is called operational morality by Wallach and Allen and implicit ethical agent by Moor. This implies that a researcher takes ethical values into account in the design of the agent. Functional morality or explicit ethical agent describes the actual implementation of ethics in the agent. At full moral agency, the agent is able to think and to act ethically like a human. Operational morality can be done through for example the use of VSD designing an ethically valid system. Functional morality is still being researched due to its difficult nature of deciding which decision is ethically valid.

WHAT WAYS ARE THERE TO RESOLVE CONFLICTS BETWEEN VALUES?

Resolving conflicts between values seems to be underexposed by researchers. VSD states that trade-offs have to be made between conflicting values, but not exactly how this should be done. Answers are rather inconclusive and trial and error seems to be the way to go.

WHAT ARE THE REQUIREMENTS FOR INCORPORATING VALUES INTO SCE?

The sCE methodology is divided into the three stages. These stages are the foundation, specification and evaluation stage. Each of these stages will be examined with values in mind. This section discusses how ethical values can be taken into account in each of these stages.

Coinciding elements of VSD in sCE can be seen in Figure 6. The conceptual investigation coincides with elements of the operational demands and human factors research, while the technical investigation is

very similar to the envisioned technology with the additional trade-off section between values and technology. The evaluation stage and empirical investigation serve a similar purpose in reviewing and prototyping with the stakeholders. Empirical investigation is also done implicitly in the foundation.

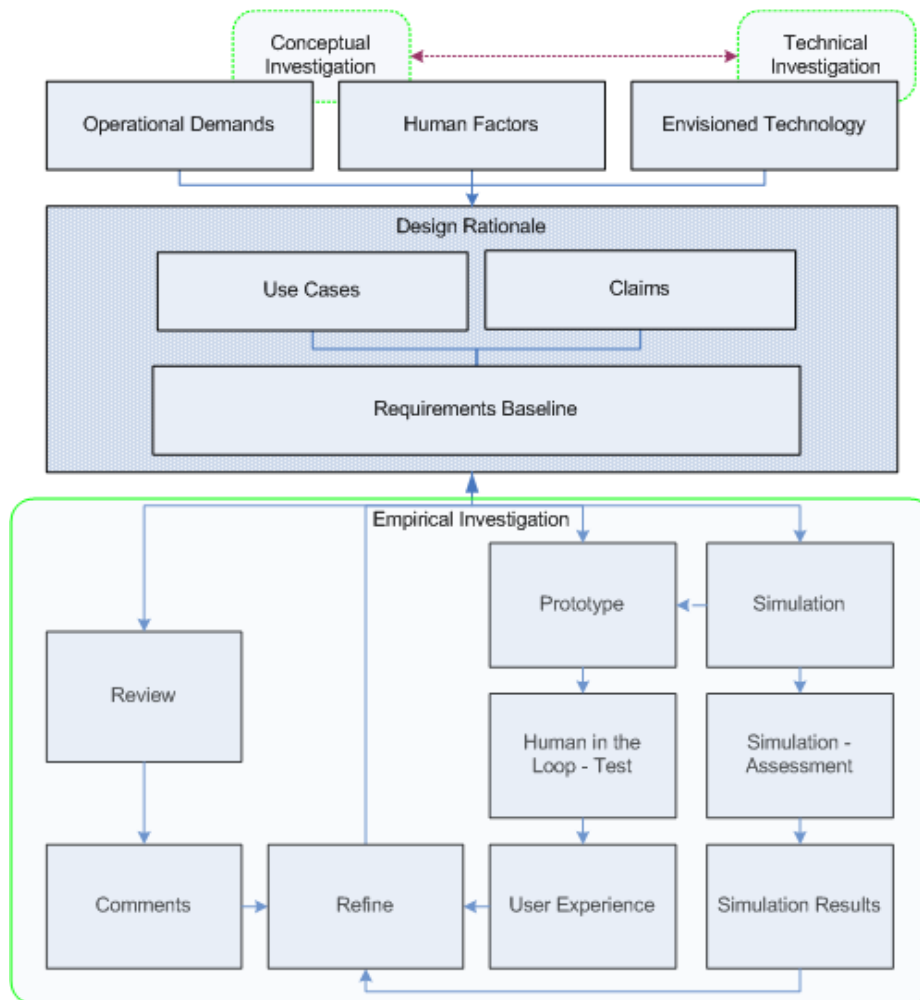


Figure 6: Coinciding elements of the investigations of VSD in sCE

The foundation stage of sCE consists of three blocks containing the operational demands, human factors and envisioned technology. Human factors might have overlap with values, but it does not explicitly mention them. It is useful to have consent on the definition of a single value, so values from a monist point of view are preferred within a project. For multiple projects, a pluralist point of view is taken in that similar moral values have different meanings in different project. So a definition of a single value is strict within a project, but might have a different meaning in another project. This would give the user a sense of flexibility. Defining values would also mean that a separate ontology, preferable for values.

Incorporating direct and indirect stakeholders from the conceptual investigation of Value Sensitive Design is useful to broaden the context of use. This would most logically be integrated in existing actors and roles study in the operational demands.

In the envisioned technology block of the foundation, values have to be taken into account similar to the approach in VSD. This would mean that the technology should be the best suited given the values and vice versa (i.e. how does the envisioned technology meets the values?). This means that there exists a reciprocity or mutual dependency between human factors and envisioned technology/technical investigation.

Next up is the specification stage in where the requirements are specified. These requirements are justified by claims and contextualized by use cases. Claims contain values and these values are not moral values, but are instrumental of nature and focus more functionality and usability. This means that there should be a separate section with claims based on intrinsic values or claims should be modified to accommodate values. Use cases are used to contextualize the functionality of the system, but ethics are leaning towards non-functionality. A way to accommodate this is by for example using misuse cases addressing the misuse or abuse of the system or more story-based methods (e.g. user stories or short fictional scenarios) to address the ethics concerns of the users.

Finally is the evaluation stage, where one or more prototypes are made and tested on a group of users. Based on this feedback, the requirements baseline is adjusted to address the issues of these users. Concluding from the VSD section, the use of qualitative evaluation methods is preferred. There should be an area to create these qualitative methods from either scratch or from templates. The overall test stage should accommodate the possibility to explicitly provide feedback to the results of the qualitative tests. Since there is no facility to specify the design, the methodology should incorporate a design baseline. The results of the qualitative tests might also be used as feedback in the design. Also, the results of the test could be linked to moral claims, which are in turn connected with the requirements.

This literature study discussed the importance of moral values in the design process. Ethics in autonomous systems will start to become increasingly important to consider, as seen in today's deployment of UAV's. This background section shows that sCE is extendable with ethical values and incorporate elements of VSD. sCE is not complete yet from the perspective of ethical values. Now the challenge is to operationalize and implement changes in the sCE tool (sCET) such that ethical values can be accounted for in a structured and rich way. This is discussed in the next chapter.

3 Design of sCEThics

In this section, the design of the new value-incorporated version of situated cognitive engineering will be discussed. This new methodology will be called sCEThics, a contraction of the sCE tool (sCET) and ethics, which is central in VSD. The design considerations of sCEThics will be according to the three stages of the SCE methodology, namely the foundation, specification and evaluation. This will be followed by an analysis of the requirements necessary for sCEThics. This section will conclude with the actual design of sCEThics.

To better illustrate the changes in the new sCEThics, a case will be used. This case is based on electronic health partners in the medical care domain. First up, the case will be briefly described at an abstract level and secondly the changes in the various sCE stages are discussed. The sCE stages are foundation, specification, and evaluation.

The main reason to utilize a case is to illustrate the difference between the theory of the literature and actual use of the methodology in practice. The case demonstrates the importance of using ethical values in decision making, as is promoted in VSD theory. However, today's version of sCE fails to address an explicit requirement with regard to ethical values. The case is based on an actual electronic health partner project called 'ePartners that care' that runs at the Dutch research institute TNO.

3.1 Case: ePartners that care

Obtaining and preserving a healthy lifestyle needs determination and motivation. This determination and motivation comes from within the person, but also from the people in the environment. When these people are not available, there should be someone or something to motivate this person. This is where the e-health partner comes in. This electronic partner should motivate the user to stay healthy and supports the user in achieving the user's goals (e.g. losing weight, intake of medicine). When it's necessary, the agent should intervene. For example, the agent should warn the user, when to take his or her medicine.

Ethical issues might arise, when the user's situation might conflict with the agent's insistence of taking medicine. There might be a conflict with or between safety and privacy. For example, should the agent insist the user to take his medicine? Should the agent contact a family member or a friend to notify that the user has not taken his medicine for a week? The system might then forgo the privacy of the user to ensure his health and safety. When is an intervention a rightful and ethical decision for the system to take?

3.2 Foundation

The foundation is divided in the three sections of the sCE methodology. These sections consist of operational demands, human factors and envisioned technology.

3.2.1 Operational demands

The operational demands contain the contextual details, including the context of use, actors, processes, tasks and roles (sCE). Parts of conceptual investigation (VSD) describe the context with the stakeholders and values within the used environment and are therefore included in the operational demands.

Stakeholders (VSD) are very similar to the actors (sCE) having an additional factor of being directly or indirectly affected by the product. To personify a stakeholder, personas can be used. This in turn would help give face to an actor in a scenario. Therefore personas are included in the specification stage to support scenarios.

Because the operational demands describe the context, it is chosen to incorporate legislation and regulations as policies into this part. This addition could be justified by the connection between ethical values and laws described in the background. A policy would describe the laws and restriction by several (indirect) stakeholders like the government, the justice department, a company or by a local instance i.e. neighborhood.

3.2.2 Human factors

Human factors contain relevant theories, principles, data and methods about the users that use the artifact. It thus concerns the theoretical understanding of the interaction between users and a product or system. While operational demands are more external to the user, human factors are more concerned with the user self.

A case is made to include the data of the empirical investigation (VSD) in the human factors part of sCE mainly because of this includes information about theories and principals. The data from research could also be linked to elements of the operational demands like context of use, tasks and roles. Statistical data from quantitative research might be better suited in the human factors section, while qualitative data like interviews might say more about the context for the operation demands. There is no clear distinction in categorizing data to one of the sections. It is better to include the empirical investigation as a separate section, because it is not always clear to which section the acquired data belongs. Therefore a new section within the foundation has been made to include all empirical investigation and other elements of VSD.

3.2.3 Envisioned technology

This part highlights the technical constraints or opportunities of a product and its benefits in comparison to current technology or other possible near-future technologies. This is also part of the technical investigation of value sensitive design. In the envisioned technology (sCE), the preference of the user is also taken into account. The technical investigation (VSD) goes a step beyond the preferences of the users in reviewing the implications of technology on the values. This connection is also served vice-versa with finding technology using values.

3.2.4 Additional requirements

These additions do not fit within the three sections of the foundation and include the empirical investigation of VSD, values elicitation and value layering.

The foundation stage addresses what should be investigated, but not how it should be investigated. Therefore both the methodology and the tool should have functionality to facilitate empirical investigation with qualitative and quantitative research types like surveys, interviews or photo elicitation. As a start stakeholder interviews and group discussions are proposed. Like discussed above,

The additional VSD elements including the empirical investigation and ethical values will be part of a separate block in the foundation.

With multiple stakeholders and several definitions of a single value, it is useful to find consensus among the stakeholders regarding the definition of a single value. Within a single project, this requires a glossary to guard consistent use of a value. Values are not independent of each other, but are most likely linked based on instrumental value or conflicting interests and as such we want to link these. Ontologies are used in information science to represent knowledge and relationship between concepts. Examples are available for geopolitics⁴ and lexicons⁵. Ontologies can be represented in the form of a graph. The graph structure makes for an easy breadth-first search for connections between two data points. Note that ontologies are not glossaries, but glossaries are in a way contained in the ontology.

The ontology can also be used over projects. There would be a specific ontology for the project, but there would also be an overall ontology with information from multiple projects. The reuse of elements in other projects is seen as low priority.

Definitions of values are most of the time rather vague. The true meaning of these values is defined by the use and context of the system. This meaning is defined and refined over time with the context becoming clearer. The new methodology should address this dynamic meaning of a value. This could for example be done by having a dictionary definition at face value and an operationalized definition combined with empirical backup like snippets from interviews or scenarios at a deeper layer. This could perhaps be an extension of the value ontology. A possible way is to layer the values with the intrinsic values at the top like a tree and the instrumental values as leaves of that tree. This should be done in a hierarchical tree structure. Every new layer will be a further specification of the value above it divided in one or more instrumental values. This tree structure is a specialized form of the graph structure of an ontology.

3.3 Specification stage

The specification stage contains four main elements. These main elements are use cases, scenarios, claims and requirements. The relationship between these main elements and other supporting elements (ontology and metrics) are illustrated in Figure 7. Requirements are at the top of the pyramid and illustrate *what* should be in the new system. Use cases are used to illustrate *how* a requirement is implemented in the system on a high level and *when* the requirement is relevant in the new system to be addressed. Scenarios give context *where* these use cases play out. Claims justify *why* a requirement is important and which effects a requirements would have on the user. Measurement of an effect is measured by a metric underpinning a claim. Ontologies are used to underpin certain terms used in use cases, claims and requirements. Specific common terms are recorded and explained in an ontology to avoid ambiguity of a meaning across developers and stakeholders in a project.

⁴ *FAO Country Profiles*, <http://www.fao.org/> Last visited on 15-11-2012

⁵ *Lexical ontologies for legal information sharing*, <http://cordis.europa.eu/> Last visited on 15-11-2012

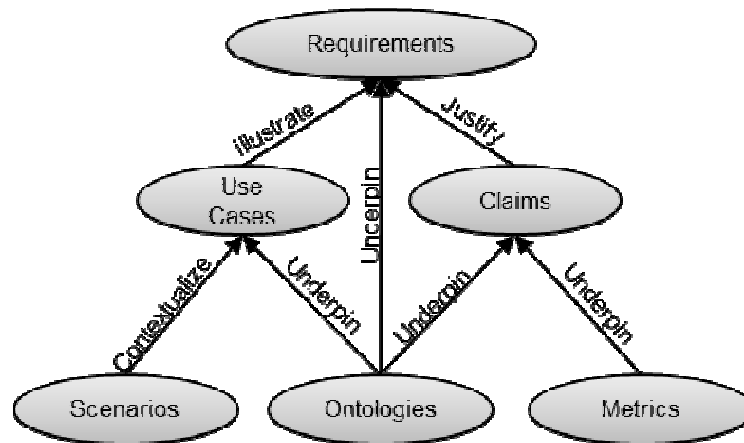


Figure 7: Relationship between the elements of the specification according to sCE

3.3.1 Use cases and scenarios

Scenarios are methods to give context to the use of the system, while use cases address how and when the system should be used. Addressing values creates a problem in using use cases to illustrate a positive or negative effect on the user. Use cases are functional of nature describing how the user uses the program and what action the system takes. Values are not functional of nature and needs to be addressed in a different way.

We believe therefore that scenarios are better tailored for addressing ethical values. A misuse case might be a way to illustrate unwanted use of the application and might apply to a few values like privacy or safety. For illustrating a bad reaction, short fictional scenarios or user stories from interviews with users might be applied. This implies that scenarios should offer flexibility to the user to illustrate the impact the system has on a user's values. By writing scenarios, specific values can be identified that are specific to that context or situation. These situated or contextualized values are harder to find without the context illustrated in the form of stories from users and might not be identified without empirical investigation.

The main difficulty was finding the additional step between the investigations of VSD and the requirements baseline of sCE. Values might have been identified, but this does not say anything about the implementation and how these values are implicated in the specification yet.

Situated or contextualized ethical values and requirements are found through conceptual investigation by the researchers with further empirical research with the stakeholders. This knowledge should be visible in the qualitative research with these stakeholders in for example an interview or a table discussion. As said earlier, this empirical investigation should be done in a separate section in the foundation. Using these high-level requirements or core functions with the values of the stakeholders and the context of a scenario, requirements can be derived. The core functions and values can then be operationalized into requirements. These requirements would need claims for justification, which could also come from qualitative research with the stakeholders.

Benyon’s scenario based design method (Benyon & Macaulay, 2002) as seen in Figure 8 can be used as a bridge between the two methodologies to facilitate the steps between value elicitation and requirements elicitation. User scenarios or stories describe the activities of the users and their possible wishes for the system and can be derived from interviews with the stakeholders. These scenarios are used to illustrate a possible problem or point of interest. Conceptual scenarios are used to make these user stories abstract and are used to generate (high level) requirements for a new system or product. Concrete scenarios are used to show the usage of a prototype of a new system within the context. A concrete scenario can be used to show off the non-functional impact of a new system. Use cases are used to formally describe the interaction of the user with the system for a concrete scenario.

Alternatively, policies can also be used to find requirements from values. By the legal nature of policies, laws and restrictions can be very strict and are sometimes already written in the way of requirements. For example, a policy from the Dutch government is to for websites to ask for permission to use tracking-cookies or privacy policies within the health domain preventing the unauthorized access of patient data by third parties. Therefore the scenario based design method of Benyon and policies are included in the new methodology.

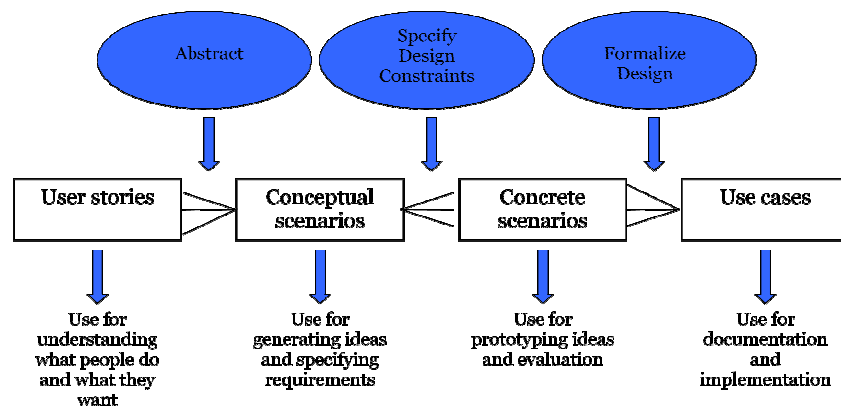


Figure 8: Four-stage scenario based design method by Benyon

3.3.2 Claims

Claims are used as a justification for the requirements and use specific metrics to measure the intended functionality. A continuous non-ethical value like performance level can be used as a metric, but is harder to measure like the instrumental ethical value well-being of an elderly person. This value is related to an intrinsic ethical value like human welfare. This means that there should be functionality to facilitate claims that are ethical of nature. These moral claims are considered either a subset or a new type of claims. A subset is more likely, because most of the functionality of a standard claims coincides with a moral claim, except for implication of the value. A way to address this is to specifically state the type of measurement for a metric like a nominal or interval scale.

3.3.3 Requirements

Requirements are linked to use cases for contextualization and linked to claims for justification. The new methodology should have flexibility for the usage of additional subsets of scenarios and claims. In the case of Benyon's scenario system, it should be possible to link a requirement with a concrete scenario. A requirement should also be linked to an ethical value.

3.3.4 Core functions

The core functions are high level requirements the system has to contain. These functions will come from the main users (direct stakeholders). In a value-sensitive environment, this might not be enough to illustrate which ethical issues might arise and the core functions should be changed accordingly. Stakeholders might also have some input for the functionality of the product. The stakeholders' input has already been (partially) addressed in the foundation with value derivation and the definition of stakeholders and their values.

Core functions and requirements should lead to a (high-level) design of the system. This design section should be an additional block to the specification stage. The content and level of the design should be up to the user. A method to address requirements is to use design patterns or solutions. (Jones, D., Stewart, S. and Power, 1999) While requirements tell something about *what* should be in the implementation, a design pattern focuses on *how* the implementation should look like. Design patterns are used to find one or more solutions for a specific problem and find a rationale behind the solution. These problems are related to one or more requirements. The rationale is related to the claims, where positive claims can be maximized and negative claims can be minimized with the right solution.

3.3.5 Additional requirements

These additional requirements do not specifically fit into the existing elements of the specification and are therefore discussed separately. These additions include a change history and visualization options.

An addition should be to explicitly mention the changes throughout the design process. A change history would help facilitate this. It's not clear yet if this should be done manually or automatically. Because the implementation of a change history could need drastic changes to the implementation, this will be researched as a point of interest for future work on the methodology and the tool.

What became clear after working out the e-partner example case, was a quickly increasing amount of data. Seeking for connections would mean that the reader should go back and forth through the document to find these connections. For example, matrices, graphs or charts would help clarify. It would be helpful to get some kind of overview about the connections between the different sections. This is more out of a practical stance for actual use rather than for the theory behind the new methodology. The overview options should work on either local (check the connections between two or three different elements) or global (get an overview of a large section of the project) level. There should be an overview option for each of the two types.

3.4 Evaluation stage

The evaluation stage is used to test the requirements to the design of the system and to refine the requirements baseline to further improve the system. The empirical investigation of value sensitive

design has some overlap with this stage. Both use human-in-the-loop and evaluation methods to refine the system in a user centered way. However, the test and refine elements of the evaluation stage describe what should be tested, but not the methods to do the tests. This could be improved by explicitly using quantitative and qualitative evaluation methods with an emphasis on qualitative, because ethical values are hard to measure quantitatively. A way to do this is to have a few readily made method templates.

3.5 Requirements

Table 3 summarizes the requirements that were discussed in the previous chapters, combined with claims. The follow requirements have come up from the design.

Nr	Requirement	Positive claims	Negative claims
1	The new methodology should incorporate the conceptual investigation of Value Sensitive Design in the foundation of Situated Cognitive Engineering	More options	Daunting
1a	It should be possible for the user to add, edit and delete values, ethical or otherwise.	Increase understanding of stakeholder's needs	Extra work, not always useful
1b	It should be possible to layer the values to further specify a value within the context.	Better adapted to context	Hard to find layered values
1c	The user should be able to express the type (direct or indirect) of a stakeholder.	Better understanding of playing field	Indirect stakeholders not always useful
2	The new methodology should include the technical investigation of VSD within the foundation of sCE.	Increase understanding in technology's opportunities & strengths	Extra work
2a	The new methodology should facilitate Trade-offs between technology and values.	Increase understanding in ethical possibilities of technology	Trade-off might not always be useful
3	The new methodology should facilitate empirical investigation (VSD) for values and stakeholders.	Increase understanding of stakeholder's needs	Extra work
3a	The user should be able to include qualitative research for the foundation including interviews and group discussions.	Increase understanding of stakeholder's needs	Extra work
4	The new methodology should include the possibility to add, edit and delete policies.	Increase understanding of restrictions	Extra work, hard to find and read
4a	Policies should be linked to values and requirements	Increase understanding of restrictions	Extra work

5	The new methodology should include a four-stage scenario system (Benyon) to facilitate the use of ethical values into the requirements.	Increase completeness	Sharp increase in extra work
5a	The four-stage scenario system should include user scenarios, conceptual scenarios, concrete scenarios and use cases.	Increase completeness	Extra work
5b	The methodology should allow flexibility in linking types of scenario to each other and to requirements to give the user less restrictions and more options.	Increase flexibility	Decreased inclination to use system
6	The claims section should be modified to address the level of measurability of a claim (Usage of scales i.e. nominal, ordinal, rational)	Increased flexibility, more options	Less clear about measurement procedure
7	The methodology should facilitate the actual implementation/design of a project.	Increase understanding of implementation	Risk of thinking too low level
7a	The design section should include option to address multiple solutions including the rationale behind a solution.	Increase understanding of implementation	Risk of thinking too low level
8	The new methodology should facilitate empirical investigation (VSD) in the evaluation stage	Increase understanding of stakeholders' needs	Extra work
8a	The user should have the option to write down an interview, a group discussion or a user walkthrough.	Increase understanding of stakeholders' needs	Extra work
8b	This empirical investigation (VSD) should be linked to the requirements, use cases and claims in the specification.	Increase structured approach	Which links & how to link not always clear
9	There should be one or more options to visualize a project within the tool.	Better overview	Possible clutter and information overload

Table 3: The requirements for the sCETHics methodology

The requirements are divided in two sections: requirements from VSD and requirements improving the work flow of sCETHics. The requirements straight from VSD include the possibility of adding, editing, deleting and layering values (req. 1a & 1b), addressing stakeholders (req. 1c), trade-offs between values and technology (req. 2) and facilitating (additional) empirical investigation in the foundation and evaluation (req. 3 & 8). Other requirements used to improve the workflow of sCETHics, include adding, editing and deleting policies (req. 4), an extended scenario system (req. 5), more flexibility in the claims (req. 6), finding solutions for requirements (req. 7) and visualization options for overview purposes (req. 9).

All requirements include positive and negative claims with the implicated effect of the requirement on the user. For the addition of values, indirect stakeholders and empirical investigation hold that the positive effect would be an increase in the understanding of the stakeholders' needs and possible

opportunities and strengths, but would lead to additional work. Policies would lead to an increased understanding which restrictions would apply on the current project, but finding these policies might not be easy and these documents are not always easy to read. The scenario system should lead to an increase in completeness and understanding, but could lead to a large amount of extra work due to dividing current scenarios into user, conceptual and concrete scenarios and creating the possibility to write extra scenarios for the sake of completeness. Design patterns should find multiple solutions to one or more requirements, but could lead to the user already thinking on a low implementation level having difficulty to focus on a higher level. The visualization options should increase the overview within a project, but could become cluttered for a large project.

3.6 Conclusion

Based on all proposed requirements the following design methodology in Figure 9 has been designed. The figure is an extended version of Figure 6. It incorporates aspects of value sensitive design, which are shown in green. The red changes are new elements that are found through the shortcomings of sCE.

The new methodology sCEThics in Figure 9 shows that sCE and VSD complement each other nicely. As stated earlier in the background, some sections of the methodologies are overlapping, like the empirical investigation with the evaluation loop of sCE and the technical investigation with the envisioned technology block in the foundation. The scenario system would in theory nicely fit in to bridge the gap between the value elicitation of VSD and requirements elicitation of sCE. The design patterns should help find solutions and thereby help understand the implementation of the prototype better. The visualization options are not included in this model, but will be included in the implementation in sCET.

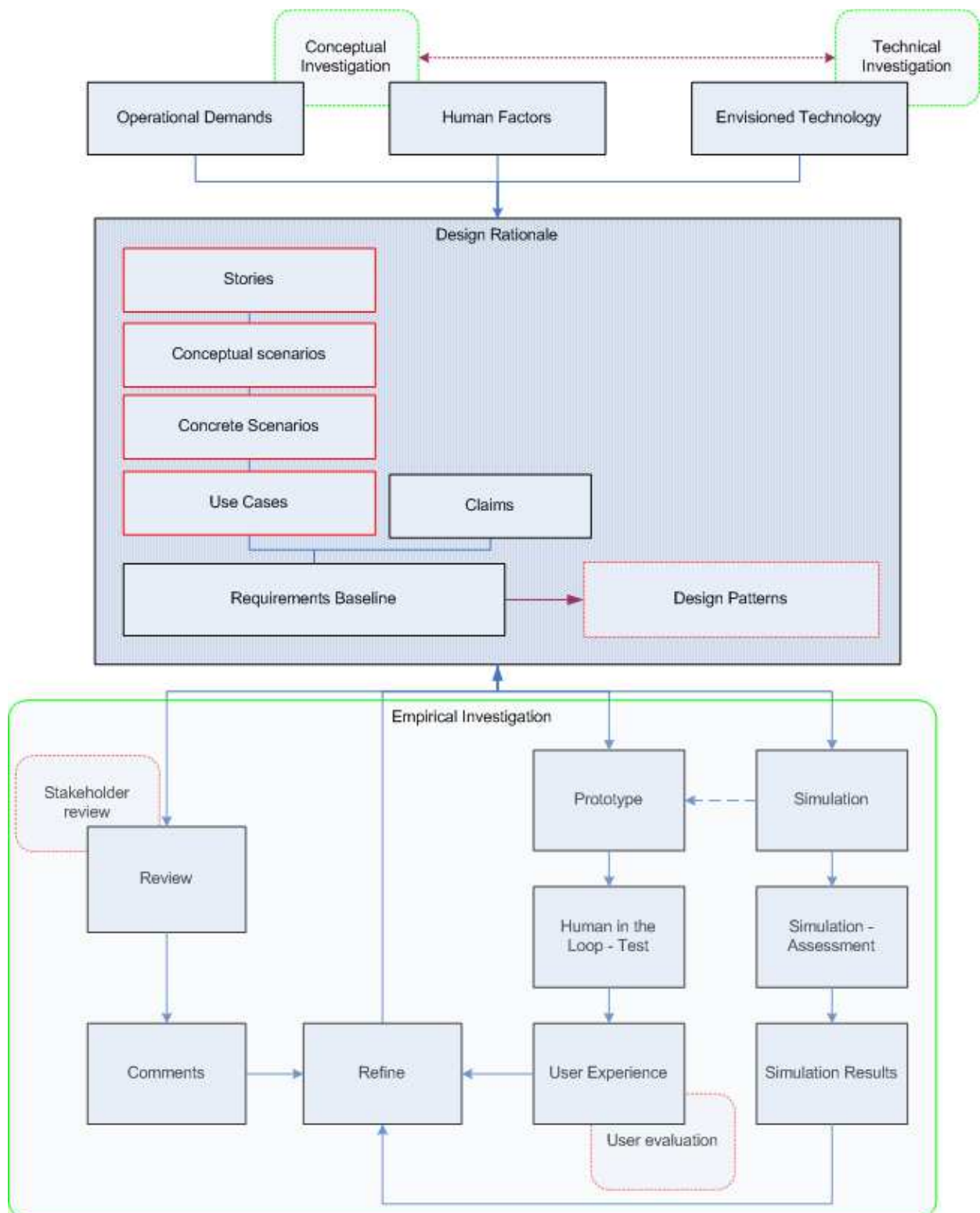


Figure 9: The new value-incorporated SCE design methodology

4 Implementation

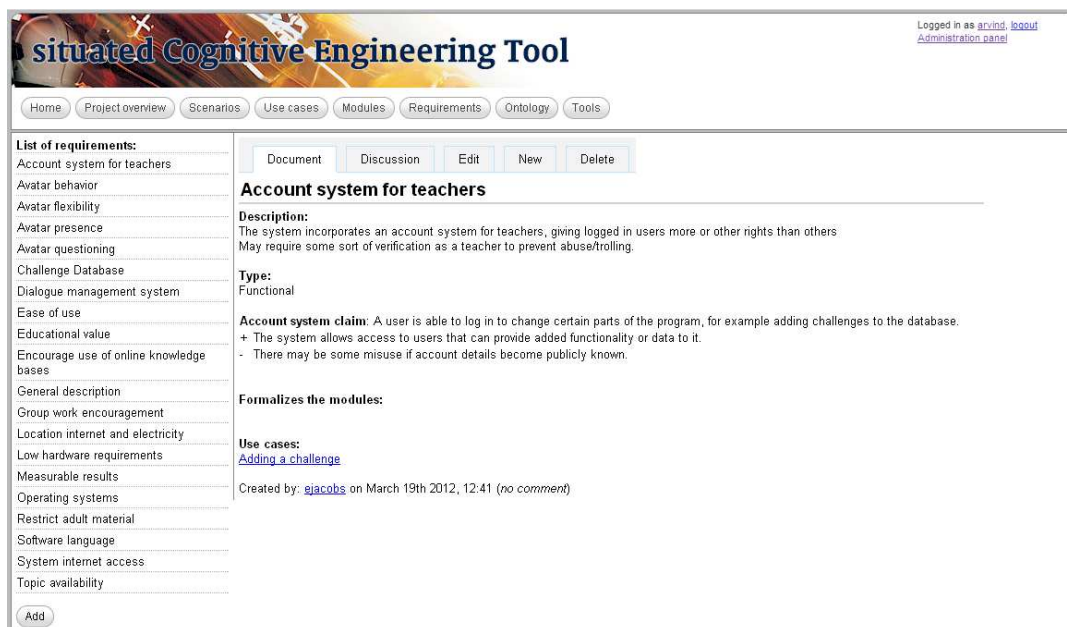
The previous chapters discussed the design of the methodology, including requirements and potential design solutions. This chapter covers important implementation considerations for the tool using the sCEThics methodology. The history of existing versions of sCET will be described in order to show the developments up to date. After that, the core of sCE is discussed followed by describing the structure of the modular design will be described. Next up, the used implementation of the tool for the tougher requirements will be explained. Finally there will be a few concluding remarks about the process of implementing the changes.

4.1 SCET History ~ Different versions of sCET

Multiple versions of the Situated Cognitive Engineering Tool have been made over the last few years. The first tool was made by Wytse Jan Posthumus as a master thesis in 2011. The second one has been designed by Jens Eldering as part of his graduation project on ePartners in conjunction with one of the frontrunners of sCET, Olivier Blanson Henkemans. The most recent one is a further development of Jens Eldering's tool by Ruud van den Beukel.

4.1.1 Earliest version of sCET ~ sCET prototype

The first iteration of the tool was made by Wytse Jan Posthumus as a prototype. It was used at TNO of the start of this thesis and is used for the 2012 Intelligent User Experience Engineering (IUXE) course in the Media & Knowledge Engineering master at the TU Delft. Figure 10 shows one of the projects done during the IUXE course and gives a general idea of the implementation.



The screenshot displays the 'situated Cognitive Engineering Tool' web interface. At the top, there is a navigation menu with buttons for Home, Project overview, Scenarios, Use cases, Modules, Requirements, Ontology, and Tools. The main content area is divided into two columns. The left column contains a 'List of requirements' with a scrollable list of items such as 'Account system for teachers', 'Avatar behavior', 'Avatar flexibility', 'Avatar presence', 'Avatar questioning', 'Challenge Database', 'Dialogue management system', 'Ease of use', 'Educational value', 'Encourage use of online knowledge bases', 'General description', 'Group work encouragement', 'Location internet and electricity', 'Low hardware requirements', 'Measurable results', 'Operating systems', 'Restrict adult material', 'Software language', 'System internet access', and 'Topic availability'. The right column shows a detailed view of the 'Account system for teachers' requirement. It includes a 'Description' section with the text: 'The system incorporates an account system for teachers, giving logged in users more or other rights than others. May require some sort of verification as a teacher to prevent abuse/trolling.' Below this is a 'Type' section labeled 'Functional'. An 'Account system claim' section states: 'A user is able to log in to change certain parts of the program, for example adding challenges to the database.' This is followed by two bullet points: '+ The system allows access to users that can provide added functionality or data to it.' and '- There may be some misuse if account details become publicly known.' There is also a 'Formalizes the modules:' section which is currently empty. A 'Use cases:' section contains a link to 'Adding a challenge'. At the bottom of the requirement view, it says 'Created by: [ejacobs](#) on March 19th 2012, 12:41 (no comment)'. The top right corner of the interface shows 'Logged in as [arvind_bogout](#) Administration panel'.

Figure 10: sCET by Wytse Jan Posthumus

This version of the tool focused significantly on the specification phase of sCE. All basic aspects of the specification phase of sCE are present in the earliest version of the sCET tool. Linking different elements like requirements and use cases is already possible in this prototype. The underlying data structure of

the sCET prototype is build upon a MySQL database with connection tables. The graphical interface is provided by PHP and JavaScript. An ontology is used to specify terms used within a project consistently. Both the foundation and evaluation are not included, which means that there is a gap in the tool to be filled. The underlying code did not allow for much flexibility. During the sCET project, this tool has been replaced by a new version that is described below.

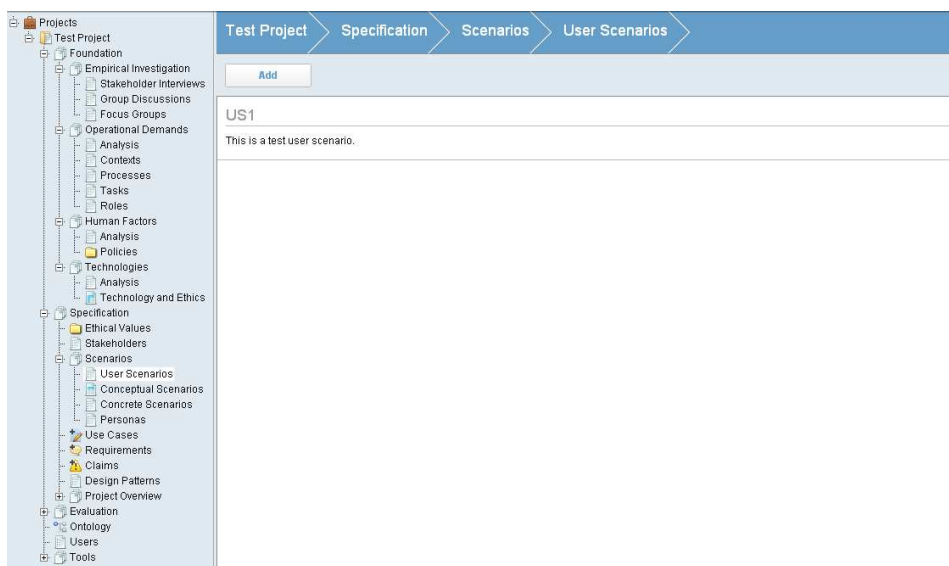
4.1.2 Fundament for a new version of sCET ~ sCET 1.0

This version of sCET could be better described as a technical redesign of the sCET prototype rather than a new iteration. It has been developed by web developer and TNO intern Jens Eldering as part of his graduation project on ePartners in the medical care domain in conjunction with Olivier Blanson Henkemans, one of the leading members of the sCET project. This version is called sCET 1.0 by the lead designers on sCET.

The core is a stripped version of a content management system called the H4 framework⁶ developed by Jens' company ATECmedia. Like the previous version of sCET, the interface is coded in PHP and Javascript with an underlying data structure in the form of a MySQL database. Communication is done using AJAX requests to send queries to and receive queries from the database. It should be easily installable on any up-to-date combination of Apache, PHP and MySQL. Unlike the previous version, this tool uses a modular design, which makes for easy customatization. These modules are defined as installable setup-files. A disadvantage of these setups comes from the restrictions in the format of these files. The modular structure of the setups will be explained later on.

4.1.3 Current version of sCET ~ sCET 2.0

The tool has been further developed at TNO by Ruud van den Beukel because certain functionality was still missing (described below) from both previous versions of sCET and the staff at TNO was still working with the sCET prototype.



⁶ H4 Framework, <http://www.atecmedia.com/>, Last visited on 17-10-2012

Figure 11: Layout of the current version of sCET

Requirement for further development included importing the old data to the new tool, the reuse of elements from other projects and a hierarchical structure to elements of the specification section. Better privacy options were added to shield off certain sections of a project for specific groups of users. More flexibility was added in the modules for additional functionality without breaking the underlying code. Another added functionality was the use of groups to add or remove groups of modules from a project. This gives the user the control to customize the layout of a project. Figure 11 shows the layout of the currently running version of sCET.

4.1.4 Advantages & disadvantages

Table 4 contains the most important advantages and disadvantages of the used versions. It shows the progress of sCET over the different implementation. Each iteration has added a new level of flexibility for both the developers (modular design) and users (more modules to use). Flexibility comes at the price of increased complexity in the code. This might mean that it can be daunting for a new developer to use the code. This does not necessarily have to mean that the system has become bloated.

Version	Advantages	Disadvantages
sCET Prototype	<ul style="list-style-type: none"> • Simple lightweight implementation 	<ul style="list-style-type: none"> • Only addresses specification • Low flexibility (strict format) • Not used anymore at TNO
sCET V1.0	<ul style="list-style-type: none"> • Powerful framework • Moderate flexibility • Easy introduction of new modules 	<ul style="list-style-type: none"> • Restricted to format of the setups of modules • Stripped CMS could contain unwanted functionality (large amount of code) • Connection system is limited
sCET V2.0	<ul style="list-style-type: none"> • Less restrictions • Slightly cleaned up code • Possible integration of research into version at TNO 	<ul style="list-style-type: none"> • Still in development (possible integration problems) • Same connection system • Low flexibility

Table 4: Advantages and disadvantages of the available versions

Because the current tool is still in development, it could lead to integration problems due to concurrent work being done. A new tool from scratch specifically for this project was discussed, but was ultimately dismissed, because of time constraints and possible integration of the newly developed modules into into TNO's system.

4.2 The core of sCEThics ~ leading to new opportunities

Before continuing on to the changes and additions, let's first look at the core of the sCET 2.0 version currently used at TNO. The core modules contain the foundation, the specification phase, an implementation of the evaluation.

The foundation aims to establish the contextual details of the new product and consists of three sections: operational demands, human factors and envisioned technology. All three sections do not

contain any subtasks, but an analysis of each section can be written as a large chunk of text. Part of the reasoning behind these decisions is that the foundation is already defined and known to the users. The main section for the user should be in the specification.

The specification phase aims to establish a requirements baseline based on the foundation with contextualization from scenarios, illustration from use cases and justification from claims; hence the tool incorporates actors, scenarios, use cases, requirements and claims. The actors were not yet included in the specification at an earlier stage of development, but were part of a module group within the foundation. The reason for this change was that important information leading up to the requirements should be included in specification. Actors supply important information related to the requirements.

The evaluation phase is used to evaluate the prototype with users and to find necessary refinements to improve the requirements baseline. Results can be documented as a single large chunk of text in the results subtask.

Ontologies can be used within this tool to specifically address certain terms introduced in the text. These terms will then be highlighted and it is possible to go over them with the mouse to show the definition of that term. Actors are highlighted in a similar way. The advantage of using an ontology is a clear use of definitions within the used context for all members within a project. The underpinning of requirements, claims and use cases is done by specifying certain terms within a project for clarification purposes.

Finally a set of tools is included for user and file management, object reuse and document export. The reuse module supports reusing requirements, claims, use cases and ontologies from other projects. The module is adjustable to facilitate other items to be reused.

Claims have an additional element of metrics. As stated in Figure 7, metrics are used to underpin a claim. A claim shows what effect a requirement has on the user. A metric facilitates the measurability of this effect.

There are two additional groups of modules that can be included: core functions and intervention mapping used at TNO Leiden for research on the medical care domain.

The first group adds a core function module to the specification section, which contains the core functions of the new system. Core functions can be seen as high-level requirements of a new system. These can be desires from a client, user or any other stakeholder.

The intervention mapping group of modules is more expansive and contains mostly modules for the foundation. It is important to note that intervention mapping is not part of sCE, but can be used to further specify elements of the foundation of the methodology. The operational demands are expanded with contexts, roles, tasks, processes and specific intervention mapping modules like problem scenarios and performance-related issues. Additional modules to human factors include determinants and theories. Envisioned technology includes the ability to compare different present and future technology with each other.

Most of these modules are not (directly) related to intervention mapping and could be seen as a more specific variant of the foundation. An option would be to split up this module group into two sections with the intervention mapping modules in one group and the additional foundation modules in another.

An important item to note is that connections are hardcoded into the modules. It would be advisable to not use connections to other modules in additional module groups. One way to avoid this would be to fully revamp the connection system, but this could cause integration problems with TNO's running system. This might be a future recommendation for the next iteration of sCET.

4.3 Modular system of sCET

The modular system of sCET 2.0 will be explained before continuing on with the implementation of the requirements. The design of sCET is structured in a tree-like way such that it is easy to include, edit and discard certain sections of a project. For example, a user comments that the envisioned technology of the foundation is not extensive enough and requests a new preference list for each developer concerning the adaptability of multiple technologies. One of the designers of sCET does not have to overhaul the structure of a project, but only needs to add a new leaf (module) to the tree (project). The modularity works on two different levels. At the first level is the design of a module as seen in Figure 12. The second level contains the design of a project setup containing multiple modules. Each of these modules is saved as a setup.php to be read by the system.

A module consists of a module type, a module name, a table name and fields. The module type, module name and table name have function as a header whereas the fields would contain the content of the module. The module type defines the view of a module. A module can be displayed as a single item, a list of items, a table or a matrix. Special types used for the project design level are groups to group modules at a design level and folders extending a group with the possibility to describe additional information about the group like for example a summary. Module and table names are straightforward. A module name describes what name to use for a module. The table name specifies the name of the module for the table in the database. The fields describe the content of a module for the user to input. A title needs to be specified for each field, except for the top header (h1). The type determines the functionality of that field.

Possible type options are simple operations like strings, text paragraphs and numbers, but also the ability to link to other modules (connect-type) and to attach files (files-type). Additional options are possible to for example address the length of a string, the size of a text paragraph and other style options. An example of such a module setup can be seen in Figure 12.

A project setup containing multiple modules needs to be installed before actual use of the system. The install will use the project setup and corresponding module setups to create a table structure in the database. Additionally, the install creates files and database entries concerning database connection, administrator access, folder restrictions against unwarranted access from outside and server access to important libraries used for the functionality of sCET.

In sCET 1.0, the module setups are used to display the fields using a view-function. A stand-alone mode has been build in sCET 2.0 to ignore the strict viewing options of sCET 1.0 with the possibility to write

more complex modules (like the radial visualization described later on). By adding a type *standalone* with a value of 1 (true) assigned to a module, the view-function would ignore the fields and use a new module.php within the module-folder instead.

```

$aSetup = array (
  'type' => 'list',
  'name' => 'Theory and Models',
  'table' => 'theories',
  'preview' => array('title'=>'name','intro'=>'intro','text'=>'text'),
  'optional' => 1,
  'title' => 'name',
  'fields' => array (
    'moduletitle' => array (
      'type' => 'h1',
      'value' => 'Theory and Models'
    ),
    'name' => array (
      'title' => 'Name',
      'type' => 'string',
      'required' => 1
    ),
    'intro' => array (
      'title' => 'Summary',
      'type' => 'text',
      'render' => 0,
      'style' => 'height:50px;'
    ),
    'text' => array (
      'title' => 'Description',
      'type' => 'text',
      'style' => 'height:200px;',
      'render' => 2
    ),
    'attachtitle' => array (
      'type' => 'h1',
      'value' => 'Attachment'
    ),
    'file' => array (
      'title' => 'File',
      'type' => 'file'
    )
  )
)

```

Figure 12: Design of a module

Of the requirements stated in the design, most of them can be done within the module-structure or using stand-alone code from sCET 2.0 like the tree map used for the requirements, which makes it possible to layer requirements upon other requirements. The overviews need more customized work. This is all summarized in Table 5.

Possible within module structure		Possible with tree map structure	Specialized stand-alone code
- Empirical investigation	- Claims	- Values	- Matrix overview
- Technical investigation	- Theories	- Policies	- Radial visualization
- Scenario system	- Design Patterns		
- Stakeholders			

Table 5: Implementation of requirements within the structure of sCET 2.0

The implementation of the modules within the module and tree map structure is mostly trivial and therefore not described in this report. The more difficult implementation of the overview items is described below.

4.4 Adding overview to sCET

The following additions do not necessarily address the new methodology, but are made to increase the overview within a project for the user. Connections are now only seen for individual items in the modules and it is not possible to quickly see an overview of (a portion of) all connections within a project. The saying goes that a picture is worth a thousand words. This should be the power of data visualization, if done right. The matrix view should enhance the local overview with the view between interconnected modules, while the radial visualization should increase the global overview for a project.

4.4.1 Matrix overview

A matrix overview is used to show an overview of the connections of claims, scenarios and use cases with two other modules. In this case, the other modules are the requirements and ethical values. The matrix is automatically generated by the system by checking the underlying database. As stated earlier in the design chapter, a positive claim would be an increased overview of the connections between the modules. A negative claim would be a danger of information overload. With every addition of a requirement or ethical value, the amount of rows or columns would increase.

The matrix view is automatically generated, while accessing the database for items and connections. The method used is quite slow, because it needs to do a query for each possible combination. For example in the matrix in Figure 13, all possible combinations of requirements and values need to be addressed to complete the matrix. For a small set of combination of ten requirements and ten values, it could already take up to fifteen seconds. This could be problematic for a large set. A possible option would be to combine multiple requests from the code to the database to speed up the process.

Requirements X Values	Control fit for user cognitive capabilities	Privacy Awareness	Limited Storage	Use of personal data	Authentication	Electronic Signature	Showing progress
Privacy control				- Abusing personal information for personal gain		- Abusing personal information for personal gain	
Privacy				- Abusing personal information for personal gain		- Abusing personal information for personal gain	
Consent				- Checking up on the patient by family or friends	- Checking up on the patient by family or friends	- Checking up on the patient by family or friends	
Accessibility				- Using personal information for profit gains - Checking up on the patient by family or friends	- Checking up on the patient by family or friends	- Checking up on the patient by family or friends	
Data Longevity							

Figure 13: Example of a matrix view containing scenarios and use cases

4.4.2 Radial Visualization

The radial visualization should increase the overview of a project for the users. This visualization was made using a powerful JavaScript library called Data-Driven Documents (D3.js in short). This library is one of the more well-known libraries used to visualize data and has been used by for example NU.nl during the European Championship Football in 2012.⁷ An example of the implementation in sCET can be seen in Figure 14.

The goal of the visualization is to show the connections between modules in the project and also to indirectly indicate the progress made on a project. The connections are shown by the visual links between the nodes and the progress indication should be visible in the slices of the pie chart. An extra ability is at the end of the page, where it is possible to select which modules should be shown in the visualization. As already stated in the design, the increased overview of a project should be a positive claim and the possibility of information overload or clutter should be a negative claim.

Creating a radial visualization means using a combination of PHP, Javascript, MySQL, AJAX requests and the data interchange format, JSON. JSON has a similar function as XML to send over information in a human-readable format. The reason to use multiple languages is inherent to the nature of the languages. While PHP is a server-side language, Javascript handles all client-side operations. When opening the web page with the visualization, all tasks related to queries on the database are done at the server-side using PHP and the visualization is done at the client side using Javascript. With the option to select modules to show, AJAX requests are necessary to facilitate the connection between the client-side Javascript code and the server-side PHP code.

The data flow should help explain the implementation. After a selection of modules for the visualization to show, Javascript code sends an AJAX request to a server-side PHP page, which in its turn contacts the database to request the items of the modules and possible connections to other modules. After all queries on the database are been done, a function in the PHP page converts the item and connection information into a JSON string. In an earlier version, the JSON string was saved to a file on the server for d3.js to use. This intermediate step was later removed by directly feeding the string to the Javascript code. The JSON string is send back over the AJAX request to the client-side Javascript code to create the visualization. D3.js uses the HTML structure to add a new scalable vector graphics (SVG) element to the body of a page to insert the visualization into the page.

⁷ *Bayern Munchen hofleverancier op EK*, <http://www.nu.nl/>, Last visited on: 10-10-2012

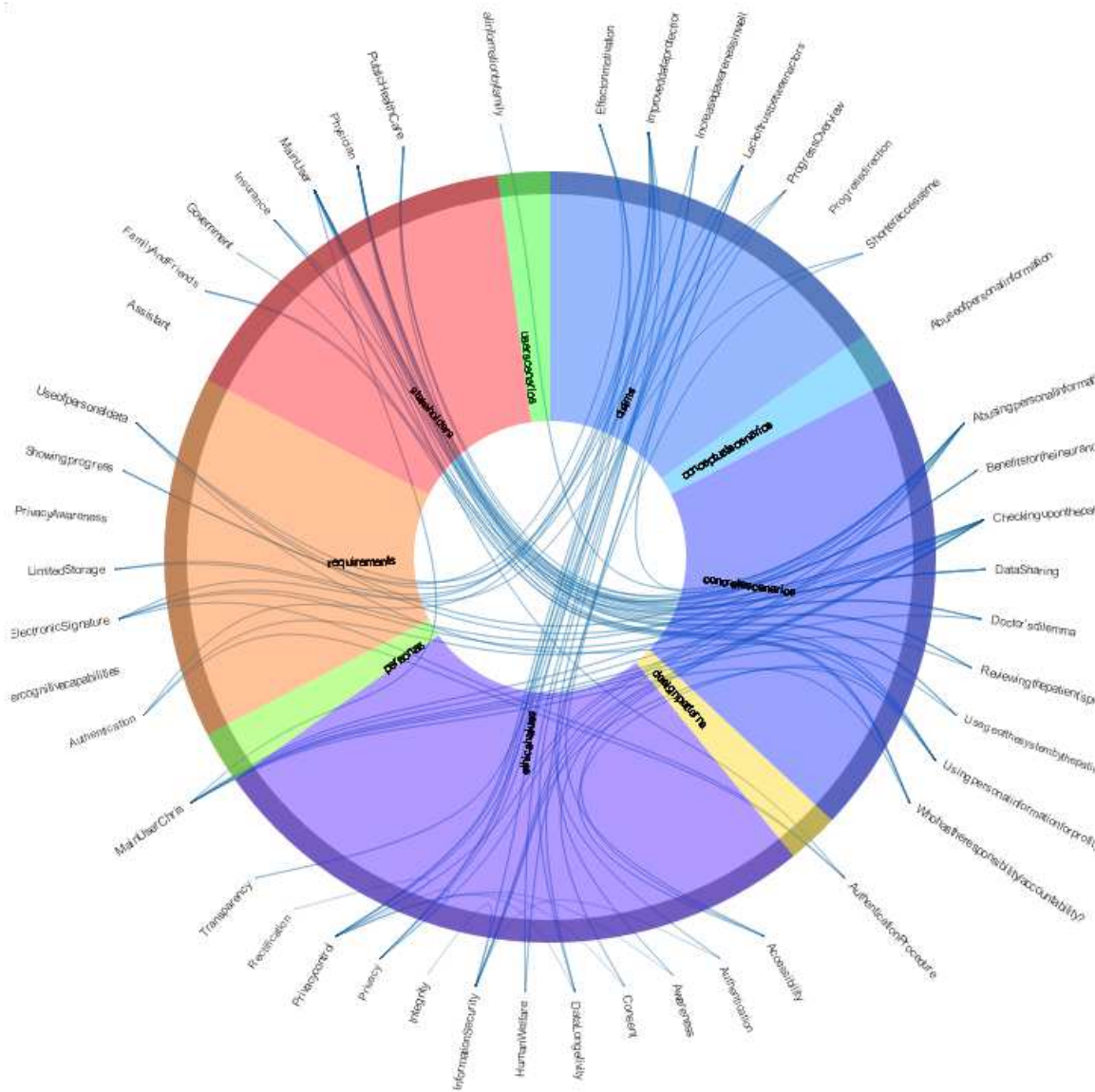


Figure 14: Example of a radial visualization

4.5 Discussion

The ups and downs of the implementation will be explained here. This includes learning how to use PHP and D3.js and concurrently working on the same code.

The first step was to learn how to use PHP, because I did not have any experience in the language. I have done a few website projects before in HTML and ASP.NET, so I did have some experience in web development. PHP may have started out as a script language, but it is possible to develop using an object-oriented method from PHP5 on. Positively, the syntax of PHP is easy to learn and uses a loose form of type checking. On the downside, it uses a loose form of type checking, causing problems about what the actual content within an object. This might not be that much of a problem working from scratch, but this is problematic for expanding an existing system.

The existing sCE code was reasonably structured, but had thousands of lines of code without actual comments. It was sometimes difficult to follow which functions are related and how they work together. Another problem at the start of the implementation was lack of flexibility to add custom code, while still staying compatible with the code at TNO. Luckily, Ruud at TNO had implemented a way to work around the existing code and made it possible to run custom code for a module with a small modification.

The Data-Driven Documents library or D3.js is a very powerful library for data visualization, but the downside of the library was the steep learning curve. This took a large amount of trial and error to create a nice visualization. A small mistake would cause the visualization to not show up. The radial visualization is based on the code of one of the examples on the website. The problem with the code was the use of attributes not included in the parsed text or file, but somehow made by the library. This unexpected behavior is still now very clear to me.

The implementation was done concurrently with the 2.0 version. Positively, I could continue working on what I was doing and there was the possibility of integrating some modules in the TNO version. Problems with working concurrently arise, when a new version of Ruud was online and had to be integrated into my version. These problems were not always clear to me and I sometimes had to consult Ruud on how to fix these. Ruud was helpful enough for these problems to be fixed quickly.

There were some problems in installing the tool on the server. This was mostly due to a lack of documentation of the tool. It was easy to use the tool locally using a WAMP⁸ called EasyPHP, but there were problems getting the tool running on a better known WAMP called XAMPP and on both a Windows server and a Linux server at the TU Delft. The problems were related to a text replacing PHP module called mod_replace and a functionality in the sCET code called OneJs gluing all Javascript files into a single Javascript file for speed purposes. Most of these problems were fixed in cooperation with the system administrator and both sCET developers.

It is important to note that the implementation is done within the boundaries of the module structure of sCET 2.0 and should not be considered as a perfect implementation of the methodology. To do so, more time and possibly access to change the core of sCET would be necessary. A few of the possible changes are already discussed in the design section. More possible future work is described and tested in the evaluation and results sections and is discussed in more detail in the conclusion.

⁸ Acronym for the combination of Windows, Apache, MySQL and PHP

5 Evaluation

This chapter describes the evaluation of the new tool for the sCEThics methodology. First, the goal of the evaluation will be explained. Next, the type of evaluation and the participants will be described. Then finally the procedure including the participants will be discussed. This chapter purely states the evaluation procedure. The results and the discussion of the results will follow in the next chapters.

5.1 Goal of evaluation

The goal of the evaluation is to evaluate the requirements of the new methodology using the sCEThics tool. Situated Cognitive Engineering is being used at TNO for research in the human factors and health domain. Because of sensitive and ethical information, ethical values should be taken into account in the design of these projects. The requirements on one hand address elements of VSD and on the other hand should give the users a (better) overview of their projects. The usefulness and (partially) ease-of-use of the additions are measures to test the sCEThics tool supporting the new methodology, which combines VSD and sCE.

Usefulness is described as the added value of specific elements that are introduced to a system, hopefully better serving the workflow of the user⁹. Another way to describe it is to find out if the changes in the current system are necessary and how useful these changes are. Although the focus of this study related to improving sCET with elements taken from VSD leading to a better design process, the ease-of-use is not a priority. However it can possibly be tested at the same time without much additional costs in order to determine potential improvement related to easy-of-use.

5.2 Literature on testing usefulness

The Technology Acceptance Model (TAM) is a widely known theory for information systems to find out in what capacity users will accept and use new technology. There have been two iterations of the model. The first iteration was by Davis in 1989 (Davis, 1989) and the second one was done by Davis in collaboration with Venkatesh (Venkatesh & Davis, 2000).

The first iteration is shown in Figure 15 and contains the perceived usefulness (PU) and perceived ease of use (PE) as variables that influence the attitude toward and intention of use to a new system. The technology Acceptance Model 2 has an additional layer with variables that influence the PU and PE containing experience, voluntariness, image, job relevance, output quality and result demonstrability. There are proposals for a third iteration of the model with an additional layer with even more variables to test (Venkatesh & Bala, 2008).

The still relevant main variables in the original model, shown in Figure 15, are the perceived usefulness and the perceived ease of use. New iterations of the model also have these two variables at its core with layers of additional external variables on top of the original model.

⁹ *Beyond usability testing: Assessing the usefulness of your design*, <http://www.slideshare.net/Banderlin/>
Last visited on 18-10-2012

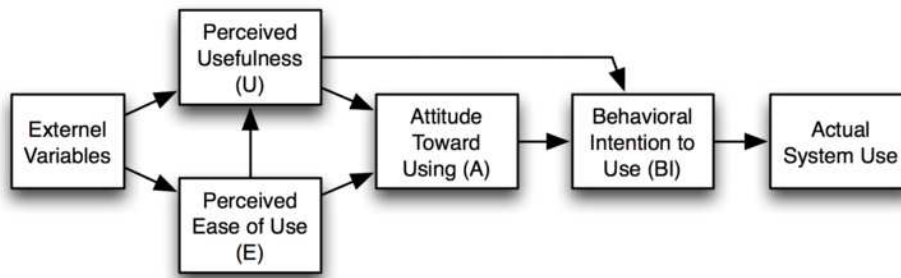


Figure 15: First iteration of the Technology Acceptance Model (Davis, 1989)

Testing usefulness is a field that has limitedly been exploited. As far as literature goes, there is hardly any information to be found of specifically testing the usefulness rather than the usability or ease-of-use. A presentation by consultancy firm Mad*Pow shows how to use usability testing techniques for testing usefulness. On suggestion of one of the professors, the repertory grid method has been examined. This method is used to find out why certain elements are important to a participant. This method was discarded, due to difficulty in combining the method with the actual use of the tool.

Mad*Pow's way of describing usefulness is to find mental models and core values of the users, so both the usefulness of the tool and the user experience can be improved. The testing consists of three phases: Pre-task, task and post-task. Pre-task questions are used to find the mindset and expectations of the user. This is important because these expectations can be used as a baseline for this user. The tasks are necessary to find out the link between implementation and expectation and results in finding the satisfaction level of the user towards the tool. The goal of post-task questions is to address the implementation and find out opportunity for improvement.

Usability testing is done has been done using many methods. Like, for example, the card sorting method; alternatively, one of the more common ways to test ease of use is by use of a think aloud session for the tasks with the help of standardized questionnaires.

5.3 Type of evaluation

This evaluation will be a qualitative research. The evaluation will, following the method of the Mad*Pow usefulness lecture¹⁰, use the three-phase model containing pre-task, task and post-task. The pre-task and post-task phases will consist of formally structured interviews. The tasks consist of six tasks related to the implemented modules, which in their turn are based on the requirements.

The evaluation evaluates dependent variables for each task being the usefulness and the positive and negative claims of the requirements as described in Table 6.

¹⁰ UPA Boston 2011 – Beyond Usability Testing, <http://vimeo.com/29425129>, Last visited on 18-10-2012

5.4 Procedure for evaluation

The evaluation will begin with an explanation of both the research and the goal of the experiment followed by a short introduction on either sCE or VSD, depended on which of the two methodologies is not yet known to the participant. The evaluation leader will ask about the experience in VSD and sCE (how well do they now the concept? Are they teaching it? Are they using it to design products?) Also, it needs to be recorded what the level of the participant is (senior researcher, prof, etc). In short, background information of the participant is important. These questions will be part of the pre-task questions.

The following questions are asked during the pre-task questionnaire. All questions use a scale of 1 for none to 5 for maximum, unless noted.

1. How much experience do you have with User Centered Design (UCD)?
2. How much experience do you have with using values in UCD projects?
3. How much experience do you have with Situated Cognitive Engineering (sCE)?
4. **Open question:** In what way have you used sCE in practice?
5. How much experience do you have with Situated Cognitive Engineering Tool (sCET)?
6. **Open question:** If yes, what are some advantages and disadvantages of sCET?
7. How much experience do you have with Value Sensitive Design (VSD)?
8. **Open question:** If yes, in what way have you used VSD in practice?

Any additional comments are remarked in open text paragraphs at the end of each question. After the experience questionnaire, any missing information on sCE and VSD will be given and the sCEThics methodology will be explained.

Following the experience questionnaire and further information on VSD, sCE and sCEThics, a set of tasks will be done to find in what way expectation and reality are comparable, thereby testing claims. Because of a lack of time, the users will not fill in an actual case. It will be a walkthrough with the methodology in mind with the user thinking out loud based on the task-specific activities. In Table 6, six tasks have been laid out according to the tested element. Also, the expected positive and negative claims for each task have been laid out. While doing these tasks, participants will need to note or say aloud their positive and negative remarks.

The following questions will be asked for each task:

- **Open question:** In what way could this element be useful to you?
- **Open question:** How would this element help and hinder you in your regular design workflow?
- On a scale of 1 to 5, how much do you agree with (positive claim)?
- On a scale of 1 to 5, how much do you agree with (negative claim)?
- **Open question:** How could it be improved?

Requirement	Task description	Positive Claim	Negative Claim
Adding and layering ethical values	Experiment leader shows an example of layered values within a test project based on ePartners that care. User can examine module and remark or ask question.	Increases understanding of needs	Extra work, might not always be as useful
Policies in the form of laws and restrictions	Experiment leader shows an example of policies within a test project based on ePartners that care. User can examine module and remark or ask question.	Increase in understanding restriction	Decrease in creativity
Scenarios & interviews to bridge gap between sCE and VSD	Experiment leader explains Benyon's system with the model on paper and shows an example of scenario system within a test project based on ePartners that care. User can examine module and remark or ask question.	Increase in completeness	Might increase workload
Design Patterns to bridge the gap between requirements and design	Experiment leader explains idea behind design patterns with only examples of design patterns and shows an example of design patterns within a test project based on ePartners that care. User can examine module and remark or ask question.	Understand implementation of the requirements	maybe too practical of mind
Claims & Scenario overview as an overview option	Experiment leader shows an example of both matrix views within a test project based on ePartners that care. User can examine module and remark or ask question.	Increase overview	Possible information overload and lack of detail
Radial Visualization as an overview option	Experiment leader shows an example of the radial visualization within a test project based on ePartners that care. User can examine module and remark or ask question.	Increase overview	Possible information overload and lack of detail

Table 6: Tasks for the participants

Finally post-task questions will be done to find out if the implementation agrees with the expectation of the participants and which improvements have to be made to address any concerns based on the implementation. The following questions will be asked:

- In what way do the expectations and the implementation in the tool agree?
- **Open question:** How could the tested elements be improved to better agree with your expectations?
- **Open question:** What is still missing from the tool?

- **Open question:** What are possible further recommendations?
- **Open question:** How do you think the system can help to design as a distributed team? What would you additionally need?

Optionally there will be an extra set of questions to test a few hypotheses of my own about future research. For each of the hypotheses, the participant can give a score between 1 for a very bad idea and 5 for a very good idea. Any other remarks can be written down in a text paragraph after each question. The following elements will be explained:

- Allow more flexibility in sCET for the user to add other connections.
- Add an additional visualization in the form of a graph to have a different overview.
- Design the system in a graphical way like a graph as an alternative to the standard interface.
- Add a logging system to sCET to see what every user has been doing with possibility of revision control.
- **Multiple choice question:** How should the logging be done?
 - Automatic
 - Manual
 - Mixture of automatic and manual
 - Other suggestion
- **Open question:** How should the logging system look like within sCET?

5.5 Participants

The candidates should either have used the sCET tool on a project base or be highly experienced using the VSD methodology with basic knowledge on using user centered design in work or research. The amount of candidates is eleven, which is low for those who fit these criteria and are available. Another handicap is that sCET 2.0 has just been official launched at TNO. A small group of employees have used the system for testing, but actual large projects have not yet been done within the tool.

The candidates will be split up in two groups. One group has experience with using both the situated cognitive engineering methodology and the tool. The participants at the human factors department at TNO Soesterberg and health care department at TNO Leiden have used sCET 2.0 in some way in their projects. The amount of experience with sCET varies between relatively extensive usage by the frontrunners of sCET and less experience for the regular employees. This group will consist of about (more than) five participants. The other group is experienced in using value sensitive design, but has not used the tool. This will be done with a smaller amount of participants of three or four persons.

It consists of one professor at the ethics and technology faculty at TU Delft, who gives a VSD course at his faculty, one professor with knowledge on both VSD and sCE and a PhD student with a subject on VSD and has general knowledge on sCE at the Interactive Intelligence section of the Computer Science department. It might be necessary to instruct both groups on the missing information.

6 Results

In this chapter, the results of the evaluation will be discussed. First up will be the outcome of the pre-task questionnaire followed by describing the results of the task questionnaires, which contain an element of usefulness and of claims. Next up, the results of the post-task questionnaire will be examined and finally will be the results of the future work questionnaire. The average scores for each claim tested in the tasks have been taken. The average is taken over 11 participants with a minimum score of 1 for not useful at all and a maximum score of 5 meaning very useful.

6.1 Pre-task questionnaire

The results of the pre-task questionnaire can be seen in Table 7.

Question	Mean	Median	Std. Dev.
How much experience do you have with User Centered Design (UCD)?	4.00	4	1.15
How much experience do you have with using values in UCD projects?	2.23	1	1.17
How much experience do you have with Situated Cognitive Engineering (sCE)?	3.69	4	1.49
How much experience do you have with Situated Cognitive Engineering Tool (sCET)?	3.38	4	1.66
How much experience do you have with Value Sensitive Design (VSD)?	1.92	2	1.32

Table 7: Results of the experience questionnaire showing the score mean, median and standard deviation. (min 1, max 5)

The pre-task questionnaire confirms the participant profile. Most of the participants have at least heard of the term UCD and have done projects using the basic principle, but have not really used values in these projects. Experience with sCE seems to be more than average, while experience with the tool is a bit less. This is explainable since the tool has just been taken into production. Participants have heard about VSD, but have never used it in practice. There seems to be a negative correlation between experience with sCE and experience with VSD.

6.2 Task questionnaires

6.2.1 Grading usefulness

Table 8 contains a summary of the results from the usefulness questions in the task questionnaires. Because the usefulness was not graded on a scale, the comments have to be interpreted as either positive or negative. The conclusions contain a general mark of the level of usefulness for each task based on the sentiment of the comments.

Requirement	In favor of usefulness	Against usefulness	Conclusion
Values	<ul style="list-style-type: none"> • Theory of value gives guidance • Justification of claims/requirements • layering to increase understanding • Possible starting point of project • Extra element for decision making • Get discussion between stakeholders • More user-centric approach to design 	<ul style="list-style-type: none"> • Current implementation too abstract • Extra work • Implementation lacks guidance • Longer design phase (potential benefit at later stage) • What kind of layering? • Where to start with values? • Blurry line between values and requirements 	In favor, but needs work
Policies	<ul style="list-style-type: none"> • Compliance with regulations to prevent overlook at later stage • Clear link possible between values-policies-requirements (i.e. Safety - Safety regulation policy - Safety compliance requirement for system) • Central place for everyone to see 	<ul style="list-style-type: none"> • Policies not always clear • Possibly out-of-date, while developing system • Not all, just relevant policies in • Extra work • Where to find these policies? • Potential out-of-the-box thinking (possibly not best solution) 	In favor, but needs work
Scenario system	<ul style="list-style-type: none"> • Clear distinction between types of scenario • Process is more explicit • Systematic approach to scenario building • Alternative to Intervention Mapping approach 	<ul style="list-style-type: none"> • Extra work • Not much extra gain • Mostly done during brainstorm session / more graphically • Scenarios are not backed up by empirical evidence • Not in all projects as useful 	Against, benefits do not outweigh the costs.
Design patterns	<ul style="list-style-type: none"> • Distinction between requirements and (possible) solutions • Intermediate step between requirement and prototype • More concrete than requirements • Generic design patterns usable 	<ul style="list-style-type: none"> • How well do these solutions work? (Examples, empirical evidence) • Extra work • Might be at a too detailed level • Partially already done with claims and use cases 	In favor, but needs work

Matrix overview	<ul style="list-style-type: none"> • Quick and easy overview • No extra work required • Check if most important requirements are covered 	<ul style="list-style-type: none"> • Possible clutter for large projects • Connection between ethical value, requirement and claims/scenario not always clear. • Should everything have to be filled in? 	In favor, but needs work
Radial visualization	<ul style="list-style-type: none"> • Quick and easy overview • No extra work required • Looks nice • Helpful for reports and presentations 	<ul style="list-style-type: none"> • Preference for predefined colors than random colors • No access to link • No possibility to link within visualization • Possible clutter 	Highly in favor. Might need a few small adjustments improve usability

Table 8: Summary of the task questionnaires including the pros and cons of the implemented requirements

For all tasks except the scenario system, the participants seem to be in favor. The radial visualization gave the most enthusiast reactions and just needed a few small adjustments to fully comply with the participants needs. These additions are related to the usability of the radial visualization and not the actual usefulness of it.

For the other favorable tasks, the idea behind the new additions was found helpful, but the actual implementation had a few shortcomings. The most important issue here was the lack of guidance in finding and addressing these elements like values and policies rather than to document these elements. All shortcomings are taken into consideration as possible improvements in Table 10.

Participants were very reserved with the idea of extending the scenario system. There seemed to be a disconnect between the needs of the users and the actual implementation. An often-heard comment was that the benefit of using the scenario system does not outweigh the amount of extra work necessary to completely fill in all types of scenarios. They would like to have the option to put in visual scenarios in the form of for example diagrams or storyboards. The addition of personas is seen as a positive addition, but most participants would like to have an option to add visual elements to the persona.

6.2.2 Measuring claims

The results of measuring the claims are documented in Table 9.

Requirement	Postive claim	Mean	Median	Std. Dev.	Negative claim	Mean	Median	Std. Dev.
Values	Increase understanding of needs	4.20	4	0.92	Extra work	2.82	3	1.08
Policies	Increase understanding of restrictions	3.45	4	1.04	Decreased creativity	2.09	2	1.14
Scenario system	Completeness	2.45	2	1.44	Extra work	3.64	4	0.67
Design patterns	Increased understanding of implementation	3.36	4	1.50	Increased practicality with cost of decreased creativity	2.55	3	1.44
Matrix overview	Increase overview	3.64	4	1.03	Information overload	2.18	2	0.98
Radial visualization	Increase overview	4.09	4	0.79	Information overload	1.55	1	0.80

Table 9: Results of the task questionnaires showing the score mean, median and standard deviation. (min 1, max 5)

The results show for five out of the six requirements, positive effects. The mean show the average score, while the standard deviation is used as a measure of variation. For most requirements, the results for the positive claims are on the right side of the average mean of 3. The results for the negative claims are for the most part below the average mean of 3. The exception to the rule here is the scenario system.

The most positive claim results are from the values and radial visualization requirements. Both have a high mean with a low standard deviation for the positive claims, meaning a higher than average satisfaction with a low spread. The participants did not find the visualization too demanding to have a serious effect on their information load (negative claim). The mean of the negative claim for the values requirement is relatively high, more towards the middle. The participants find it a decent amount of extra work, but not exceptionally though.

The claims for policies, design patterns and matrix visualization have a mean slightly above average for the positive claim and a slightly below average mean for the negative claim. The median for the positive claim shows that most participants are agreeing with the positive claim. The added value of all three requirements is understood, but the implementation did not entirely convince the participants.

The positive claim for the scenario system requirement has a below average mean with a high standard deviation. The negative claim has an above average mean. These results together with the usefulness remarks indicate that the extended scenario system is not a useful addition to sCE. Noted remarks and suggestions from participants made it apparent to use visual aspects like storyboards and diagrams to increase its usefulness and decrease the amount of extra work or through these changes at least spent their time in a better way.

6.2.3 Possible improvements to the implementation

Table 10 contains the conclusion of Table 8 extended with the possible improvements for the implementation of the requirements. These possible improvements were given by the participants during the task questionnaires.

Requirement	Conclusion	Possible improvements
Values	In favor, but needs work	<ul style="list-style-type: none"> • Guidance system • Reuse for other projects • Set of predefined values • Highlight tension fields/conflicts between values • Support sessions to find values, like access to standardized value questionnaires
Policies	In favor, but needs work	<ul style="list-style-type: none"> • Reuse for other projects • Sets of predefined policy categories for different types of projects • URLs to relevant policy sites • Highlight relevant parts in policy document • Highlight conflicts between policies • Search function • Examples of policy in practice • Include URL/file and date
Scenario system	Against, benefits do not outweigh the costs.	<ul style="list-style-type: none"> • Reuse for other projects, if possible • Use of audio and visuals (audio recordings, pictures, videos, diagrams) • Visual persona • Collaborative scenario writing with validation
Design patterns	In favor, but needs work	<ul style="list-style-type: none"> • Reuse for other projects • Use of tested design patterns (i.e. HCI or software design patterns) • Use of visuals (pictures, diagrams, video) • Validation (experts or examples) • Scoring of solutions • Link to claims and use cases
Matrix overview	In favor, but needs work	<ul style="list-style-type: none"> • Level of abstraction/detail for large projects • Clickable links • Easy manipulation of matrix (add/remove elements)
Radial visualization	Highly in favor. Might need a few small adjustments improve usability	<ul style="list-style-type: none"> • Grouping and coloring of elements • Clickable links • Show conflicts • Export to picture • Add and remove links within visualization • Level of abstraction/detail for large projects

Table 10: Possible improvements to the implemented requirements

The improvements for the first four requirements can be summarized in improvements for speed up, help functionality and usability. Reuse and predefined lists are functionalities that all participants are really in favor of to speed up the process and help give additional insight within a project. Most participants would like to have some form of guidance and help (e.g. URLs to relevant sites) within the tool to both speed up their work and to help them understand why this module is of importance to a project. Audio and visual options should help improve usability and the understanding among users of the tool.

Most of the improvements for the overviews are related to the usability of the implementation. Additional manipulation like adding/removing links and levels of detail would need serious work and testing.

6.3 Post-task questionnaire

Next up, the results of the post-task questionnaire will be examined. Table 11 shows the results of the post-task questionnaire, which was targeted to get an understanding of the expectancies of the participants.

Question	Mean	Median	Std. Dev.
How much did the implementation meet your expectations?	3.82	4	0.75

Table 11: Results of the post-task questionnaire showing the score mean, median and standard deviation. (min 1, max 5)

The participants did not really have expectations before, but the implementation seems to meet their expectations to a certain degree. There is still room left for improvement on the VSD additions to the sCEThics tool. Some of the participants noted that there is a risk of over-specification. The tool should be used to speed up the design process and the VSD additions should take this in mind.

As seen in Table 12, comments about missing functionality within sCET focus on workflow and usability related issues. This includes functionality like reusability, explicit conflict resolution, better usage of multimedia for scenarios and personas, special version for stakeholders, support for data collection and overall guidance within the tool.

The participants are overall positive about the usage of sCET within a distributed team. Any additions are on the subject of communication, logging options and revision control. Most participants find it important to know who has done what and why they have made a certain choice. Users should have the possibility to discuss choices and to revert items to an earlier version.

Question	Remarks
How could the tested elements be improved to better agree with your expectations?	<ul style="list-style-type: none"> - More explanation and guidance - Reuse from other projects - Facilitate research rather than documentation (i.e. links to relevant websites, templates/examples) - Improved linking to existing sCE modules

What is still missing from the tool?	<ul style="list-style-type: none"> - Capture evaluation outcome - Show/highlight conflicts - Use of audio and video (i.e. pictures, diagrams, videos)
What are possible further recommendations?	<ul style="list-style-type: none"> - Version of sCET for stakeholders to look in and fill in
What would you additionally need in sCET to help you design in a distributed team?	<ul style="list-style-type: none"> - Logging system - Version/revision control - Communication and discussion tools (i.e. forums, chatroom, who is online?) - Project progress indication

Table 12: Selected remarks on the open questions of the post-task questionnaire

6.4 Possible future work

Finally, the results of the possible future work questionnaire will be discussed. The results can be seen in Table 13 and Figure 16.

Possible future work idea	Mean	Median	Std. Dev.
Allow more flexibility in sCET for the user to add other connections.	3.10	3.5	1.29
Add an additional visualization in the form of a graph to have a different overview.	4.40	5	1.07
Design the system in a graphical way like a graph as an alternative to the standard interface.	4.22	4	0.67
Add a logging system to sCET to see what every user has been doing with possibility of revision control.	4.20	4	0.92

Table 13: Results of the future work questionnaire showing the score mean, median and standard deviation. (min 1, max 5)

The result for the future suggestion of *more flexibility* is divided among participants as seen in the standard deviation. While it would give more freedom to users, illogical connections can be made. The current implementation also serves as a form of structure and guidance to the user, which more flexibility would break.

Additional Visualization is seen as great idea. Any type of visualization is seen as a useful addition. The *graph visualization* could benefit from a level of detail option to easily switch between a general overview and a more detailed view. Graphical way of connecting elements in sCET looks like a good idea.

A *Logging system* is seen as a good idea, if done in combination with revision control. Participants could in addition choose from multiple options and the results are seen in Figure 16.

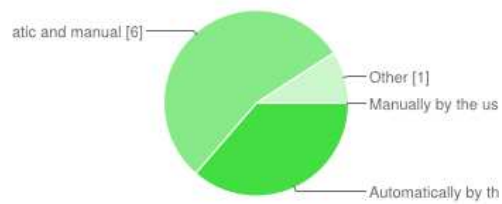


Figure 16: Results on the type of logging (manual, automatic, mixture or other solution)

Automatic seems to be the answer with possibly so help of the user. The user should then be able to add additional comments to the changes. It is not yet clear how this logging system should look like.

Suggestions range from a simple time-sorted table structure for each module to more complex systems like Apache Subversion¹¹ or Kanban¹².

¹¹ *Apache Subversion*, <http://subversion.apache.org/>, Last Visited on 12-11-2012

¹² *Kanban Applied to Software Development: from Agile to Lean*, <http://www.infoq.com/>, Last Visited on 12-11-2012

7 Discussion

In this chapter, there will be a discussion about the results of the previous chapter. Table 14 contains a summary on the ratings of the usefulness and claims of the requirements taken from the previous chapter. The scale goes from -- implying no impact on the user at all on the claim to ++ meaning the user is impacted a lot by the claim.

Requirement	Conclusion on usefulness	Positive claim	Rate	Negative claim	Rate
Values	In favor, but needs work	Increase understanding of needs	++	Extra work	+/-
Policies	In favor, but needs work	Increase understanding of restrictions	+	Decreased creativity	-
Scenario system	Against, benefits do not outweigh the costs.	Completeness	-	Extra work	+
Design patterns	In favor, but needs work	Increased understanding of implementation	+	Increased practicality with cost of decreased creativity	+/-
Matrix overview	In favor, but needs work	Increase overview	+	Information overload	-
Radial visualization	Highly in favor. Might need a few small adjustments improve usability	Increase overview	++	Information overload	--

Table 14: Conclusion on the usefulness of the requirements

7.1 Values

The participants saw the usefulness of values in theory, but they did not know where to start and how to find these values. This is understandable due to the non-trivial nature of values. Their suggestion was to give some guidance in the form of possible reuse from other projects, lists/questionnaires of values and links to websites with values. These suggestions could be part of possible future work on the sCEThics tool including the suggestion of trade-offs.

The positive claim scored as expected. The participants understood how addressing values can increase their understanding of the needs of the stakeholders and also that this would make them write better claims and requirements. The average rating of the negative claim is related to the non-triviality remark in the previous paragraph. It is not always clear where and how to use values within a project.

In the evaluation version of sCET, the ethical values module was located at the specification stage. Mostly the sCET frontrunners noted that this module should be in the foundation rather than in the specification. Their reasoning behind this remark is that the specification stage should only contain information directly related to the specification of new product or system, which is not the case when defining values and stakeholder. These are valid reasons and therefore the ethical values should either be part of the operational demands, human factors or a separate section for the VSD modules. I have

chosen for a separate section for all elements of VSD including the ethical values in the foundation, because all elements of VSD will then be together as a single package and there is no clear indication where it should be in the existing sections of the foundation.

7.2 Policies

Participants know what to document here, but not exactly how and where to find policies. The same suggestions were brought up again: reuse, suggestion lists and links to relevant sites. Like with values, which policies are relevant might not always be trivial, therefore these suggestions will also be taking into account for possible future work.

The participants understood how addressing policies would increase an understanding of the restrictions within a project. At the same time, they did remark that the language used in policy documents is not always trivial and more interpretations are possible for a policy.

The comments on constraining creativity were interesting. While laws and restrictions could be strain on the creativity, it could also bolster creativity by searching for workarounds. Some commented that there might not be a strain on their creativity, because they work on new technology where policies do not exist yet.

7.3 Scenario system

Most participants did not find the scenario system to be very useful in practice due to the extra work necessary outweighing the benefit of structure and completeness. This is similar to what was expected with the claims, but the overall outcome was more negative than positive. Like previously stated, an approach with more audio and visual options like storyboards and diagrams are more helpful to the participants and therefore should be included as future work.

The scores for both the positive and negative claims is disappointing, but completely understandable. While participants did understand the additional steps necessary to have a clearer distinction in the scenario system, they did not mind not being incomplete. As the negative claim shows, the extra work put into it would certainly outweigh the benefits in this case.

7.4 Design Patterns

The theory behind the additional step between requirements and design was understood by the participants, but they were not entirely sure whether they would use it in practice. Some suggestions were made to improve this module with real-life examples of solutions as a validation step and to use generic and tested design patterns from software design and human-computer interaction¹³. Patterns are already used in these examples; therefore they must work in practice. This seems like a good idea, because these patterns are already validated in some form and the users do not have to reinvent the wheel.

¹³ *Pattern languages in human-computer interaction*, <http://www.hcipatterns.org/>, Last visited on 07-11-2012

The implementation did slightly increase the understanding of the implementation to the participants, but not as much as expected. A possible explanation would be that the idea of what should be in a design pattern was not clear and how this was a substantial addition next to requirements, use cases and claims. On the other side, it did not make most participants think too technically. There is a quite large variation, which means that a part of the audience did think that the current implementation of the design patterns make them think too low-level.

7.5 Matrix overview

While it was clear what content is shown in the matrix overview, it was not exactly clear to the participants which relationship exists between the elements within the matrix. Due to the relationship not being entirely clear, participants stated that they would not use this overview much. A possible change would be to make the matrices more customizable, where the axis and content can be chosen.

The matrix overviews did increase the overview for the participants. A explanation is that there was no such overview in sCET yet and therefore any overview is an increase over no overview at all. The matrix structure keeps the overview clean, so the participants did not really have find the overview to give a real strain on their information load. The matrix structure might not be optimal for large projects, but can be improved with options to hide and show details.

7.6 Radial Visualization

The radial visualization was given an instant positive reaction by the participants. This is an element they had wanted to be in sCET for a while and found the module a very useful addition to the tool. There are a few small bugs left, but it is fully functional and I am proud to say that the radial visualization is now implemented in sCET 2.0 at TNO.

The positive claim of increased overview scored higher than the same positive claim for the matrix overview. The radial visualization gives an quick insight into the connections between modules within a project. There was almost no strain on the information load of the participant. For large projects, it might get crowded. Options to hide and show details could help avoid this.

7.7 Possible future work

Based on the results of the possible future work questionnaire and the remarks by the participants, the additional graph visualization, designing within the graph and a logging system with additional revision control is all seen as great additions. These additions will be pursued in possible future work. Allowing more flexibility gave more controversy and might therefore better be left alone for now.

7.8 General observations

A reoccurring theme during the tasks was reusing elements from other projects. This would save time and could also give users another perspective, which they might not have seen before. This comment does not only hold for the new requirements, but for sCET 2.0 in general.

The empirical investigation put some extra effort in the evaluation stage, but not enough according to the participants. A new iteration should focus on improving the evaluation stage with for example options for questionnaires, interviews and analysis options.

There seemed to be a discrepancy between implementation of the VSD modules within sCET 2.0 and needs of users. Participants would like more guidance in the tool next to the dry documentation of the steps. The tool should focus on explaining, directing and facilitating the user's activities rather than documentation (i.e. Examples, useful links and templates to guide the user). Using the tool is just a small part of their daily activities and is not their highest priority. When projects have to be documented within sCET, it has to be fast and easy.

8 Conclusion

This final chapter contains an overall conclusion to this master thesis. First, the main research question from the introduction will be answered. Next, the overall experience of the thesis will be explained. Subsequently, several recommendations for future work will be laid out. Finally, there will be a take away message containing the scientific contribution of my master thesis.

8.1 Research question

Systems are getting more intelligent and emergent potentially impacting ethical values. Such ethical values need to be accounted for early in design process. Therefore there should be a methodology that addresses these values systematically and shows their impact on other parts of the design in a structured way. In the introduction, the following main research question was asked:

HOW COULD SITUATED COGNITIVE ENGINEERING AND VALUE SENSITIVE DESIGN BE COMBINED TO ADDRESS VALUES IN A SYSTEMATIC MANNER?

This master thesis has taken the angle of using Value Sensitive Design to adjust Situated Cognitive Engineering to incorporate values into the design. VSD and sCE complement each other with VSD stating how and sCE stating what to address. All three investigations of VSD can be built in the foundation stage of sCE. The conceptual and empirical investigations are a separate block within the foundation, while the technical investigation is part of the envisioned technology block of sCE. Nine requirements were distilled and implemented. Six of these nine requirements were used for evaluation excluding the empirical and technical investigation.

sCEThics was evaluated with eleven participants. The results reveal that the implementation of requirements relating to values, policies and design patterns are seen as useful, but needs revision to reduce the workload for the sCEThics user. Suggestions for revision from participants include the reuse of values, policies and design patterns and linking to relevant sites to find them. The extended scenario system was not seen as very useful, due to the benefits of the model not outweighing the amount of extra work necessary. Writing down a single scenario already takes a lot of time, writing additional scenarios cause more extra work without having notable benefit. Both visualization additions were seen as useful because the overviews created a better insight to the connections within a project and also showed which connections were still missing. Neither of the two visualizations gave the participants any large strain on their information load, but both did give them a nice overview. The radial visualization was seen as more useful than the matrix overview, because the radial visualization gave the participants a better overview of the whole project at first sight and understood it without additional explanation.

Because the scenario system was too much work, something else has to be found to address the gap between values and requirements. A suggestion would be to emphasize on modeling trade-offs between conflicting values and use these trade-offs to write-down new requirements. Another possibility as suggested by the participants is to use visual instead of written scenarios.

What immediately became apparent during the evaluation is that there is a difference between theory of the methodology and the actual use in practice using the tool. While the implementation is a dry

conversion of the sCEThics methodology, the users have a slightly different need. The tool and methodology should act as a guidance and extra insight for them to build better systems. All initial investigation was done with experts on the sCE and VSD methodology and less so with actual users of the sCET tool. These requirements could and maybe even should have been identified earlier on. As a counterpoint, taking interviews with users would not have been that useful due to lack of actual use of both sCET 1.0 and sCET 2.0.

8.2 Future work

Some suggestions for future work have already been addressed in the evaluation during the future work questionnaire. Below is a list of suggestions for future work including those earlier suggestions.

- Incorporate examples and guidance system for values, policies and design patterns to guide and inspire the users to come up with a better design for the product
- Visual scenario system with the use of diagrams, images and other audio and visual options.
- Further research use on changing the matrix overview to be of added value to the users of the tool.
- Small usability improvements to radial visualization.
- Graph visualization will give an additional and different type of insight to the user
- The visualizations should add interactive elements to add and remove connections between elements.
- Add progress logging and revision control to the tool to get an insight into the progress of a project.
- Further research into Intervention Mapping (IM) to see if elements of IM can be taken into account within sCEThics.
- Put more emphasis on supporting the evaluation stage with for example the possibility to conduct questionnaires and interviews and analyse empirical investigation.

8.3 Take away message

Every year new intelligent systems are being introduced. Bringing more intelligence in sensitive domains would mean that developers and researchers should take better care of the values and needs of the stakeholders in designing an intelligent system. While there are still ways to go, the sCEThics methodology should be a first step forward in the ethical design of intelligent systems, while also taking the context into account.

The requirements elicitation and constant refinement of sCE makes for a better user experience. By adding values, the understanding of the user's needs and trade-offs between conflicts of values is also added. Early recognition of trade-offs could lead to requirements that might have been overseen without values elicitation and stakeholder analysis. By linking values elicitation with requirements elicitation, sCEThics increases the understanding of the origin of requirements from stakeholders and makes it easier to trace decisions made in the design.

9 References

- Anderson, R. E., Johnson, D. G., Gotterbarn, D., & Perrolle, J. (1993). Using the new ACM code of ethics in decision making. *Communications of the ACM*, 36(2), 98–107. doi:10.1145/151220.151231
- Arkin, R. C. (2008). Governing lethal behavior: embedding ethics in a hybrid deliberative/reactive robot architecture. *Development*, 121–128. doi:10.1145/1349822.1349839
- Benyon, D., & Macaulay, C. (2002). Scenarios and the HCI-SE design problem. *Interacting with Computers*, 14(4), 397–405. doi:10.1016/S0953-5438(02)00007-3
- Borning, A., & Muller, M. (2011). Next Steps for Value Sensitive Design. *Library*, (September).
- Cheng, A.-S., & Fleischmann, K. R. (2010). Developing a meta-inventory of human values. *Proceedings of the American Society for Information Science and Technology*, 47(1), 1–10. doi:10.1002/meet.14504701232
- Dantec, C. A. Le, Poole, E. S., & Wyche, S. P. (2009). Values as Lived Experience: Evolving Value Sensitive Design in Support of Value Discovery. (S. Greenberg, S. E. Hudson, K. Hinkley, M. RingelMorris, & D. R. Olsen, Eds.) *Design*, 1141–1150. doi:10.1145/1518701.1518875
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. (T. Ball & R. B. Jones, Eds.) *MIS Quarterly*, 13(3), 319–340. doi:10.2307/249008
- Dix, A. (2008). Response to “Sometimes it’s hard to be a robot: A call for action on the ethics of abusing artificial agents. *Interacting with Computers*, 20(3), 334–337. doi:10.1016/j.intcom.2008.02.003
- Epstein, R. G. (1994). The case of the killer robot (part 1). *SIGCAS Comput Soc*, 24(3), 20–28.
- Floridi, L., & Sanders, J. W. (2004). On the Morality of Artificial Agents. *Minds and Machines*, 14(3), 349–379. doi:10.1023/B:MIND.0000035461.63578.9d
- Friedman, B., & Kahn, P. H. (2003). Human Values, Ethics and Design. In J. Jacko & A. Sears (Eds.), *Handbook of HumanComputer Interaction* (pp. 1177–1201). Lawrence Erlbaum Associates. .
- Friedman, B., Kahn, P. H., & Borning, A. (2003). Value Sensitive Design and Information Systems. (P. Zhang & D. Galletta, Eds.) *Technology*, 3(6), 1–27. doi:10.1145/242485.242493
- Gotterbarn, D., Miller, K., & Rogerson, S. (1999). Software engineering code of ethics is approved. *Communications of the ACM*, 42(10), 102–107. doi:10.1145/317665.317682
- Jones, D., Stewart, S. and Power, L. (1999). Patterns: using proven experience to develop online learning.
- Moor, J. (2006). The nature, importance, and difficulty of machine ethics. *Intelligent Systems, IEEE*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667948

- Murphy, R., & Woods, D. D. (2009). Beyond Asimov: The Three Laws of Responsible Robotics. *IEEE Intelligent Systems*, 24(4), 14–20. doi:10.1109/MIS.2009.69
- Nagel, T. (1979). The Fragmentation of Value. *Mortal Questions* (pp. 128–141). Cambridge University Press.
- Neerincx, M. A. (2010). Situated cognitive engineering for crew support in space. *Personal and Ubiquitous Computing*, 15(5), 1–14. doi:10.1007/s00779-010-0319-3
- Neerincx, M. A., & Lindenberg, J. (2008). Situated Cognitive Engineering for complex task environments. In J. M. Schraagen, J. M. C. Militello, L. Ormerod, & R. Lipshitz (Eds.), *Naturalistic Decision Making and Macrocognition* (pp. 373–390). Aldershot, UK: Ashgate Publishing Limited.
- Norman, D. (1986). Cognitive engineering. *User centered system design*. Retrieved from http://www.personal.psu.edu/jzk170/blogs/jung_hyup_kim/Cognitive_engineering.pdf
- Poel, I. van der. (2012). Chapter 24 . Translating values into design requirements. *Philosophy*.
- Pommeranz, A., Detweiler, C., Wiggers, P., & Jonker, C. (2011). Elicitation of situated values: need for tools to help stakeholders and designers to reflect and communicate. *Ethics and Information Technology*. doi:10.1007/s10676-011-9282-6
- Schroeder, M. (2008). Value Theory. (E. N. Zalta, Ed.) *Stanford Encyclopedia of Philosophy*. The Stanford Encyclopedia of Philosophy. doi:10.1111/1467-9973.00225
- Strawser, B. J. (2010). Moral Predators: The Duty to Employ Uninhabited Aerial Vehicles. *Journal of Military Ethics*, 9(4), 342–368.
- The Economist. (2011). Flight of the drones: Why the future of air power belongs to unmanned systems.
- Venkatesh, V., & Bala, H. (2008). Technology Acceptance Model 3 and a Research Agenda on Interventions. *Decision Sciences*, 39(2), 273–315. doi:10.1111/j.1540-5915.2008.00192.x
- Venkatesh, V., & Davis, F. D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46(2), 186–204. doi:10.1287/mnsc.46.2.186.11926
- Wallach, W., & Allen, C. (2009). *Moral Machines: Teaching Robots Right from Wrong*. *Philosophical Psychology* (Vol. 25, pp. 1–5). Oxford University Press, USA. doi:10.1080/09515089.2011.583029

10 Table of figures

Figure 1: Values hierarchy by Ibo van de Poel (Poel, 2012).....	14
Figure 2: Two dimensions of Artificial Moral Agent development	16
Figure 3: Bridging the gulfs of execution and evaluation	19
Figure 4: The three conceptual models.....	19
Figure 5: Situated Cognitive Engineering.....	20
Figure 6: Coinciding elements of the investigations of VSD in sCE	23
Figure 7: Relationship between the elements of the specification according to sCE	28
Figure 8: Four-stage scenario based design method by Benyon	29
Figure 9: The new value-incorporated SCE design methodology	34
Figure 10: sCET by Wytse Jan Posthumus.....	35
Figure 11: Layout of the current version of sCET	37
Figure 12: Design of a module	40
Figure 13: Example of a matrix view containing scenarios and use cases.....	41
Figure 14: Example of a radial visualization.....	43
Figure 15: First iteration of the Technology Acceptance Model (Davis, 1989).....	46
Figure 16: Results on the type of logging (manual, automatic, mixture or other solution)	57
Figure 17: How much experience do you have with UCD?.....	73
Figure 18: How much experience do you have with using values in UCD projects?.....	73
Figure 19: How much experience do you have with Situated Cognitive Engineering (sCE)?	73
Figure 20: How much experience do you have with Situated Cognitive Engineering Tool (sCET)?.....	73
Figure 21: How much experience do you have with Value Sensitive Design (VSD)?	73
Figure 22: By addressing the values and stakeholders, you can write better claims and requirements... 74	
Figure 23: How much of a strain would this be on your workload?	74
Figure 24: How much does this increase your understanding of possible restrictions?	74
Figure 25: How much would this restrain you in your creativity?	74
Figure 26: How much do you care about the completeness of the scenario system?	74
Figure 27: How much extra work would this be for you?	74
Figure 28: How much would this make you understand the implementation of the requirements?	75
Figure 29: How much does this element force you to think in terms of low-level implementation issues rather than high-level functionality?.....	75
Figure 30: How much would this increase your overview of the connections within the project?.....	75
Figure 31: How much would the strain be on your information load?	75
Figure 32: How much would this increase your overview of the connections within the project?.....	75
Figure 33: How much would the strain be on your information load?	75
Figure 34: In what way does the implementation match your expectations?	76
Figure 35: More flexibility in connecting to other parts within a project	76
Figure 36: Visualizing the project as a graph or tree.....	76
Figure 37: Designing the system as a graph rather than a traditional interface	76
Figure 38: Adding a progress logging system to a project.....	76

11 Table of tables

Table 1: Advantages and disadvantages of VSD	19
Table 2: Advantages and disadvantages of Situated Cognitive Engineering	21
Table 3: The requirements for the sCEThics methodology.....	32
Table 4: Advantages and disadvantages of the available versions	37
Table 5: Implementation of requirements within the structure of sCET 2.0.....	40
Table 6: Tasks for the participants	48
Table 7: Results of the experience questionnaire showing the score mean, median and standard deviation. (min 1, max 5)	50
Table 8: Summary of the task questionnaires including the pros and cons of the implemented requirements	52
Table 9: Results of the task questionnaires showing the score mean, median and standard deviation. (min 1, max 5)	53
Table 10: Possible improvements to the implemented requirements	54
Table 11: Results of the post-task questionnaire showing the score mean, median and standard deviation. (min 1, max 5).....	55
Table 12: Selected remarks on the open questions of the post-task questionnaire	56
Table 13: Results of the future work questionnaire showing the score mean, median and standard deviation. (min 1, max 5)	56
Table 14: Conclusion on the usefulness of the requirements	58

12 Appendix 1: Expert reviews

12.1 Expert Interview: Ibo van de Poel

Date: April 2012 (Checked by Ibo van de Poel on May 8, 2012)

Ibo van de Poel is Professor in Ethics and Technology at the Delft University of Technology. He has been lecturing Ethics and Technology since 1997. One of his subjects is teaching Value Sensitive Design to students. His point of view is not strictly coming from computer science, but rather from philosophy and engineering in general. Over the years, he has been critical of VSD, especially on the lack of any requirements elicitation, the notion of values and the handling of conflicting values. Ibo van de Poel has been working on a book on design for values. A chapter in his upcoming book addresses the requirements elicitation by including a value hierarchy linking values to requirements as seen in Figure 1. We have discussed the methodology, the requirements elicitation, values hierarchy and the evaluation of the methodology.

Ibo has been teaching Value Sensitive Design to students for 2 years now. Rather than using it as a fully developed methodology for developing new products, he sees VSD as a way of teaching students to think about the possible limitations and consequences of their new product.

The VSD methodology developed by Friedman et al. currently lacks requirements elicitation. Ibo tries to address this using a value hierarchy with the values at the top, norms in the middle and requirements at the bottom. From an example in the chapter in his book, norms seem to be coming from laws and regulation. The problem with this approach is that these laws and regulation are not always available. Ibo's comment is that these norms can be more than just laws and regulation and might also include the demands and wishes of the stakeholders and general moral norms.

One of the more interesting discussions was about the function of a methodology. Is it possible to address every possible outcome using the methodology? Ibo's answer was that is not possible and that completeness might not even be the most important part. There will always be situations that neither the developers nor the stakeholders have addressed, but using the methodology should be used as a justification for the newly developed system. It should show the reasoning behind the decisions.

Finally, the discussion ended with a question about the validation of values in designs. There are hardly any approaches available; one approach would be to evaluate internally with stakeholders and externally with the help of experts in the field. The internal evaluation could be done using a stakeholder walkthrough, sitting with the stakeholders addressing the validity and reasoning within a step. The external expert evaluation could be done by presenting the design at a conference.

12.2 Expert interview: Jurriaan van Diggelen

Date: May 2012 (Checked by Jurriaan van Diggelen on May 16, 2012)

Jurriaan van Diggelen is a project leader and research scientist at the Dutch research institute TNO. His main expertise is on artificial intelligence and human-computer interaction. He is one of the leads on the sCET project for the development of a tool to design products with the situated cognitive engineering. This task entails maintaining control and direction of the project. His current research consists of designing an electronic partner for use in elderly homes called e-partners that care. Goal of this interview is to find his vision on designing ethical systems using sCE, to find a project to test the new methodology on and the current state of the tool sCET.

Given that Jurriaan's current research is about intelligent systems in a risky domain, it would be useful to find out his view on the usage of ethical values. After explaining value sensitive design and the new sCE methodology, there was a short discussion on which values to be used in the define stage. Universal values are useful to get an idea of the issues at hand. Local values derived from stakeholders are useful to find out what they find important and what values are missing at that stage. Neither type will give a complete overview individually, so both types of values should be incorporated in the design.

Research has been done at TNO to incorporate values into the design process. Jurriaan's view on this matter was clear. He made an example of a system with high levels of privacy. By first building the system and afterwards incorporating privacy measures, the system could essentially need a redesign to facilitate these measures. By addressing these concerns earlier on, certain procedures or modules can be fully integrated into the system and could also save time seeing those matters at an earlier stage. His views were very much consistent with the literature study. His idea of splitting up privacy in other values like on a communication and personal level was similar to breaking intrinsic values down to instrumental values. His privacy principles were similar to the norms from the Value Hierarchy paper by Ibo van de Poel.

There might be a possibility to use the e-partners that care project to test the new value-based sCE methodology. The main problem with using that project for evaluation and validation is that the project is not yet advanced enough at the time of testing. Instead of using that project, a scenario or test case could be made based on the project. If it would be possible to test this case, is not clear yet.

sCET is being further developed by a small group at TNO. The version available to me is a few weeks old already and in the mean time, a different group at TNO has made several adjustments to it. There are now two versions to continue upon. The first version is the older version, in which I would have full control to do what I please, but would probably not be used in practice. The second version is a new version to be used in practice, but with a few restrictions in place. There would be no problem in adding new modules, but adding new content to the core program should not contradict the direction of the TNO version. I will presumably be using this new TNO version of sCET.

12.3 Meeting Military Human Enhancement

Date: July 2012 (Checked by Alex Leveringhaus on July 10, 2012)

A kick-off meeting for the project on Military Human Enhancement was held in The Hague at the Institute of Global Justice. It was organized by post-doctoral researchers Dr. Tjerk De Greef from TU Delft and Dr. Alex Leveringhaus from University of Oxford in the United Kingdom. Tjerk has a computer science background, while Alex comes from a philosophical and ethical stance. The meeting was interdisciplinary with experts from domains of computer science, philosophy, ethics and law. The experts were coming from the Netherlands, United Kingdom and the United States. The project is being funded by the Dutch research institute NWO and the Dutch military and navy is involved for assistance and evaluation. The meeting will not be explained in detail here, but just the details interesting to my own research.

Military, agents and design

Just war theory was mentioned several times. In essence, a just war is an armed conflict which is deemed just or fair under certain political and philosophical criteria, which include values. It is debatable if this is even realistically possible. Assuming the just war theory, agents have to be designed to meet these criteria and values. Design methodologies like VSD will presumably help to achieve the goal of ethical military robots or agents.

A fascinating subject was the question whether a robot or agent could and should be like a human or not. There were arguments for both the inferiority and superiority of robots in contrast to humans. There is an inequality in the abilities of both entities. Humans are better in reasoning with intuition and emotion, while robots are far superior in more rational and computational abilities like processing data and calculation time.

One agreement I was happy to hear, was the agreement on taking values into account at start of a project rather than a consideration at the end of the project. This would help avoid oversight at a later stage in the project. It was not surprising to hear, but it helped nonetheless.

Confidentiality was one of the topics brought up. This should be seen in the context of sensitive information given by one of the project stakeholders. Options are to leave this out of the system or include this in the system with a form of security. Currently it is opted in the current system to just leave it out.

Presentation

The meeting was an opportunity to evaluate the system with the experts from the different disciplines. Most of the audience was not very familiar with both Value Sensitive Design and situated Cognitive Engineering. Expert on ethics and technology, Dr. Jeroen van den Hoven of TU Delft already had explained a few basic parts of the methodology. The presentation consisted of explaining the use of design methodologies, quickly describing VSD and sCE, defining the new methodology and explaining it using an example.

The explanation went well, until the example was explained. A discussion went on about the use of which values and the layering and ranking of importance of certain values. One argument was that values can be layered according to intrinsic and instrumental values with intrinsic values divided into smaller, more specific instrumental values. There did not seem to be any discussion on that. The other argument was that values cannot be ranked according to importance, because of the case of value incommensurability, the lack of comparability according to accuracy. While most of the audience agreed on this, some were still convinced that a rank was possible according to the needs of the stakeholders. Surprising to me, this argument was coming from the philosophy section.

Another point of interest was the use of indirect stakeholders and their use in the design. On the slides, insurgents were mentioned as an indirect stakeholder. Perhaps the negative intonation of the word would have the audience thinking that an insurgent is an enemy. It does not have to be an enemy and it could be seen as a civilian from my point of view. There was a slight agreement on taking civilians into account, but enemies are irrelevant in the design. There is certainly truth in that, but a humane system might also take more resilient stakeholders into account.

Policies and restrictions were not only mentioned during the presentation, but seems to come up every now and then during the meeting. Rules and regulations should be maintained. Both are important to know what is possible and acceptable within the domain. Policies can be seen as part of the derive stage or foundation of the methodology and therefore will be included in the system.

Responsibility

The first subject of the project on Military Human Enhancement will focus on responsibility. A distinction is made between responsibility and permissibility. The first term focuses on the accountability of an agent, the other focuses on the tolerability of the action. While the terms are not mutually exclusive, responsibility or accountability is the main subject here.

At first, the discussion was focused on the responsibility of the human and less on the agent. Again, the inequality between humans and robots has been discussed within the context of responsibility. A remark later on was that responsibility does not have an exclusive relationship. Every agent has some kind of responsibility, but the level of responsibility decides which agent is mainly responsible. An analogy can be made between the relationship between humans and pets and one between humans and robots.

Plans for the future

Important to me, the importance of my work to this project was discussed. My research and master thesis could be further researched and made into a paper in a journal like the Journal of Ethics and IT. This will most likely not be done by me, but by one of the post-doctorate researchers. To further follow up on the methodology, a case study can be done by other master students to evaluate and validate the methodology.

12.4 Follow-up meeting Jurriaan van Diggelen

Date: September 2012 (Checked by Jurriaan van Diggelen on September 3, 2012)

Like said earlier, Jurriaan van Diggelen is one of the leads on the sCET project at TNO and is currently working on a project called ePartners That Care for developing assistive virtual coaches for the health domain. The goal of the follow-up meeting was to show the changes made to sCET and to find an opening to evaluate the new system.

Before the meeting, Jurriaan had given access to the ePartners That Care-project in the sCET version used by TNO. The project is still in the early stages with the documentation still in progress. Most of the information was within the requirements and ontology section. I have tried to adapt this case into the new value-based version of sCET to show off the increased usefulness of the new system.

First up, was to show the changes made specifically to the methodology with an emphasis on the value sensitive design additions and the changes to the specification part of sCET. It immediately became apparent that there is difference in the theoretical and practical use of the tool. Ideally, the users would completely fill in all sections. According to Jurriaan, only the necessary or important sections are filled. The additions to the specification like the stakeholders and values should be mandatorily filled in, but other new elements like interviews, discussions or technology trade-offs could be done optionally.

Continuing with the main additions, adding policies to the foundation was evident, but it might belong in the operation demands rather than in the human factors section. The addition of putting in (ethical) values was reasonably clear, although the connection with claims and requirements through means of further specifying or splitting up values was not fully self-explanatory yet.

The expanded scenario system seemed to give the biggest amount of discussion. This was about its role with respect to ethical values and the overall usefulness. Its role was made clear by explaining that a use case is pure functional and does not take a non-functionality like ethical values into account. The overall use of the scenario system should be to explicitly show the connections with the stakeholders and their stories. Because these stories are not always in the system, the tool should be flexible enough to allow other scenarios without the need of user scenarios or stakeholder interviews.

The idea behind the overviews was clear enough, but the real highlight was the visualization. According to Jurriaan, visualizations were on their future to-do list and he was hoping to add this to a future version of sCET. There was a small caveat in that it did not allow for much flexibility in showing specific sections of the project. This could be fixed by adding check boxes or list boxes for the user to select or deselect certain aspects.

As a test case, the ePartners That Care-project might be used. Jurriaan has sent additional information to make a convincing case on why to use the new methodology and tool over the current system. The new system could then be evaluated by possibly up to 8 persons assigned by Jurriaan. An evaluation procedure needs to be found to check the usefulness and ease-of-use of the system from my side and the increased usefulness for the new product from TNO's point of view.

13 Appendix 2: Tables and figures of the evaluation

13.1 Experience Questionnaire

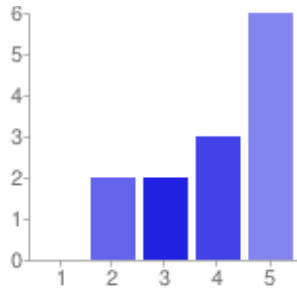


Figure 17: How much experience do you have with UCD?

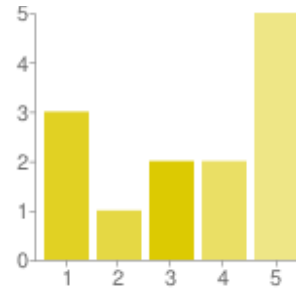


Figure 20: How much experience do you have with Situated Cognitive Engineering Tool (sCET)?

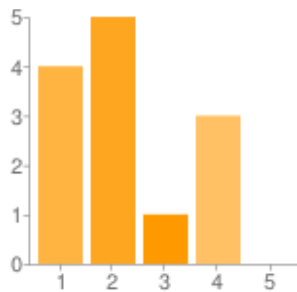


Figure 18: How much experience do you have with using values in UCD projects?

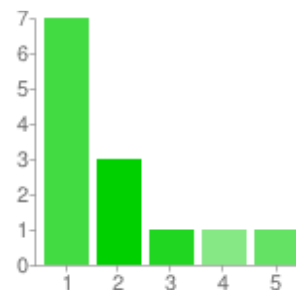


Figure 21: How much experience do you have with Value Sensitive Design (VSD)?

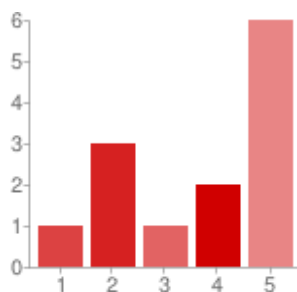


Figure 19: How much experience do you have with Situated Cognitive Engineering (SCE)?

13.2 Task Questionnaires

13.2.1 Values

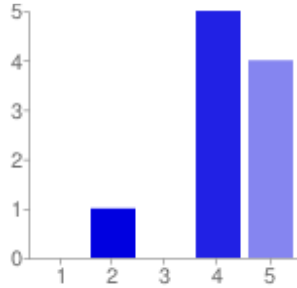


Figure 22: By addressing the values and stakeholders, you can write better claims and requirements.

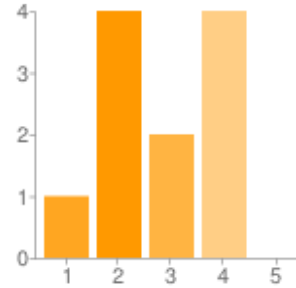


Figure 23: How much of a strain would this be on your workload?

13.2.2 Policies

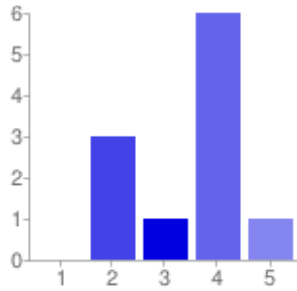


Figure 24: How much does this increase your understanding of possible restrictions?

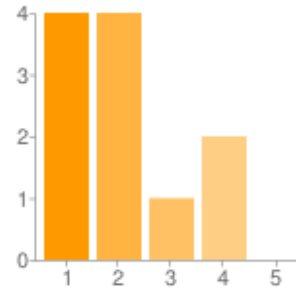


Figure 25: How much would this restrain you in your creativity?

13.2.3 Scenario system

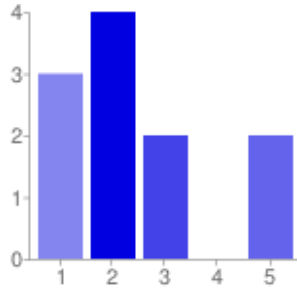


Figure 26: How much do you care about the completeness of the scenario system?

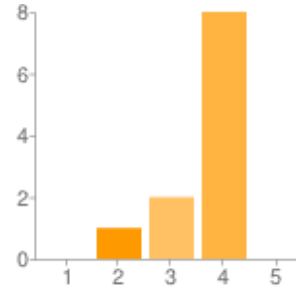


Figure 27: How much extra work would this be for you?

13.2.4 Design patterns

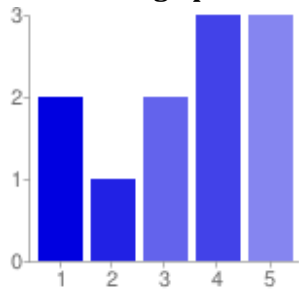


Figure 28: How much would this make you understand the implementation of the requirements?

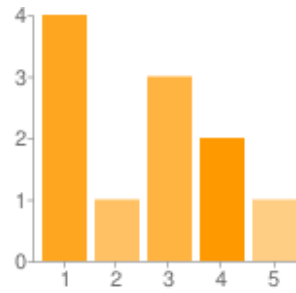


Figure 29: How much does this element force you to think in terms of low-level implementation issues rather than high-level functionality?

13.2.5 Matrix overview

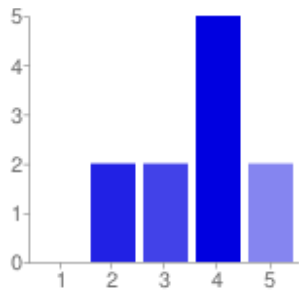


Figure 30: How much would this increase your overview of the connections within the project?

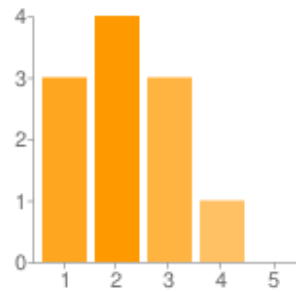


Figure 31: How much would the strain be on your information load?

13.2.6 Radial visualization

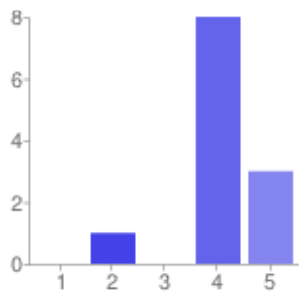


Figure 32: How much would this increase your overview of the connections within the project?

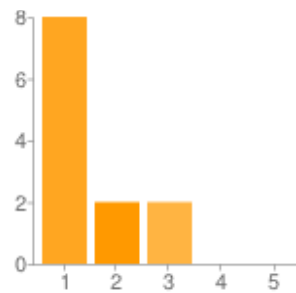


Figure 33: How much would the strain be on your information load?

13.3 Post-task questionnaire

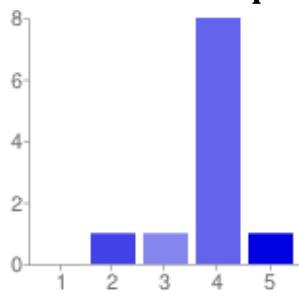


Figure 34: In what way does the implementation match your expectations?

13.4 Possible future work

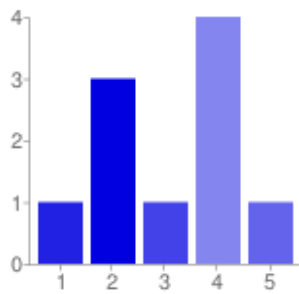


Figure 35: More flexibility in connecting to other parts within a project

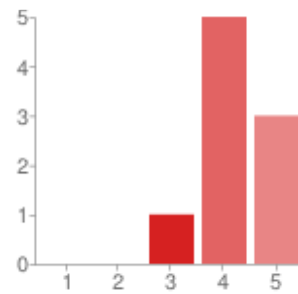


Figure 37: Designing the system as a graph rather than a traditional interface

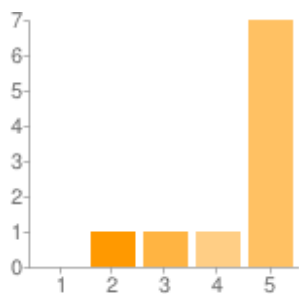


Figure 36: Visualizing the project as a graph or tree

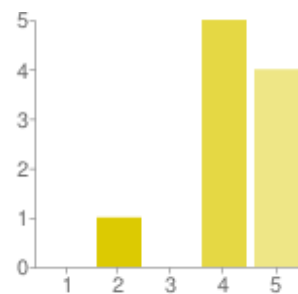


Figure 38: Adding a progress logging system to a project