Master Thesis Business Informatics

Better Design Rationale to Improve Software Design Quality

Author:

Rizkiyanto
4234634

June 2016
Final Version

Supervisors:

Dr. Floris J Bex
Dr. ir. Jan Martijn van der Werf

Universiteit Utrecht

Name: Rizkiyanto
Student Number: 4234634
Contact: _.rizkiyanto@students.uu.nl

Title Thesis: Better Design Rationale to Improve Software Design Quality
Date: June 2016

University: Utrecht University
Faculty: Information & Computer Science
Study: Business Informatics

Supervisors: Dr. Floris J Bex
Dr. ir. Jan Martijn van der Werf

Table of Content

Appendices

List of Figures

List of Tables

1 Introduction

Building software is more than just programmers creating lines of codes for a software program. There are many phases that developers have to go through to make sure the software is built according to the user expectation. A methodology called Software Development Life Cycle (SDLC) characterizes the essential stages for developers to create good software. Software Development Life Cycle describes the phases of the software development cycle and the order in which those phases are executed (Ruparelia, 2010). Each step produces deliverables required by the next stage of the life cycle. Requirement gathering and analysis translates the requirements into a design for the design phase. Then the code is created according to the design, which happens in the development phase. After coding and development, the testing phase verifies the deliverable of the implementation phase back to the requirements. In this thesis, the focus will lie on one of these phases, the design phase.

According to Dyer (1988), the design phase holds an important role since the majority of software defects (almost 60%) are introduced during the design phase and the cost of their removal can be more expensive than defects introduced at later steps in the development phase. In the design phase, the system and software design is prepared from the requirement specifications, which are gathered from the previous phase. The design phase helps in specifying hardware and system requirements and also assists in defining overall system architecture by means of design decisions.

The design decisions are made depends on the argument of the designers, the reasoning that they build during design discourse session. This reasoning process is important in software design because it could influences the quality of the design decisions and ultimately the design itself. Recent research of Tang et al. (2008) stated that the design quality could be improved with a simple design reasoning approach, especially for inexperienced designers.

To help understand the reason behind those decisions, another element of design called Design Rationale is needed. According to Lee (1997), the Design Rationale offers more than just the decisions, but also the reason behind each decision, that covers the justification, the other alternatives, and the argumentation that lead to design decisions. These additional elements offer a richer view of both the design process and the design product.

Therefore, in this research we want to focus on the elements of Design Rationale as the result of reasoning process during the design discourse. These elements could be implicit during the design process or explicitly documented in the design document.

1.1 Problem Statement

For the last decades, researchers have claimed that Rationale plays a significant role in software design. Parnas and Clements (1986) found that to document the design decision in the correct way to produce an ideal process of software design is challenging. In other words, the design documentation should be written as though the rational design process was conducted.

The design itself is a process of creativity, where the designer's idea is captured and then realized. According to Rittel and Webber (1973), design is a process of negotiation and deliberation because design involves different stakeholders from various backgrounds and requires a decision to resolve many issues. According to Dutoit et al. (2007), using a Design Rationale method improve the quality of decisions, because it can support in clarifying these issues and the trade-offs behind the design decisions. Having explicit Design Rationale would also provide an orientation for a designer to justify his design decisions. However, making design decision remains an internal thought process, and it relies on the ability of the designers to reason.

According to Tang et al. (2008) design reasoning is an important process that designers use in developing a solution. This process helps the designers to recognize design issues, find ways to design a solution to solve the issues and finally make the design decisions. If this process fails then the resulting design decision is likely to be wrong. Connecting design reasoning with Design Rationale, we could say that the design reasoning is the process for making design decisions as the product of design phase, while the Design Rationale is the justification for that product.

Previous work on Design Rationale has focused on capture and representation. Many researchers have developed approaches to capture Design Rationale such as Kunz and

Rittle (1970) with IBIS (Issue Based Information System), MacLean (1991) with QOC (Questions, Options, and Criteria), and Lee (1991) with DRL (Decision Representation Language). However, capturing or recording, design rationale is a particularly difficult problem. Recording all decisions made, as well as those rejected, can be time consuming and expensive. The more intrusive the capturing process, the more the designers resistance will be encountered. Since it is time consuming and viewed as documentation, Conklin and Yakemovic (1991) stated that Design Rationale capture is viewed as expendable if deadlines are an issue. Also, designers are reluctant to take the time to document the design decisions they did not take, or took and then rejected. According to Fischer et al. (1991), documenting the design decisions can interfere the design process if it is viewed as a separate process from constructing the artifact.

A survey of approaches to explicitly capture Design rationale by Shum and Hammond (1994) found weak evidence regarding utility and usability compared to what might have been expected given the scale of system development efforts. Since for different people to agree on a formalization scheme they must agree on the parsing, the labeling, and the linking of the information. Furthermore, the cognitive effort is very high to parsing the thoughts to fit them into a semiformal notation, and this can cause "cognitive overload". For example, when the users face a mismatch between their conception of the information and the system's formal representation. Also, people may use different terms to describe the same topic. Dutoit et al. (2007) also mentioned that cost could be the other aspect that could prevent the use of Design Rationale approaches in the software industry, capturing the rationale may add an initial cost.

While capture and representation of design rationale are important, the real value is how design reasoning can be useful for the designer and the system as well. Recent research of Tang et al. (2008) stated that the design quality could be improved with a simple design reasoning approach, especially for inexperienced designers. They conducted an empirical experiment to examine the design quality of two groups of designers, one group equipped with design reasoning and one without. To measure the quality of the design, they use the usability as the design quality attribute in the experiment. And the result of the experiment showed that by using a design reasoning approach the quality of design improves statistically.

Based on the research mentioned above, we can conclude that capturing the Design Rationale and using design reasoning affects the outcome of design decision quality. However, there has been no study, as far as we can tell, on what is the link between these two approaches.

1.2 Research Statement and Scope

The main goal of this project is to empirically explore the relationships between Design Rationale and design reasoning. In other words, we wish to better understand the relationship between the process behind the design and the design result. Therefore, the main research question is formulated as follows:

> ***RQ: What is the link between design reasoning and Design Rationale in the design process?***

In order to answer this question there are several sub-questions that need to be researched.

> ***SQ1:*** *what are the central concepts of design rationale and what is design reasoning?*

This research focuses on two approaches during the design session, the Design Rationale and the design reasoning. We first look at the central concept of these two approaches and build our conceptual model base on these findings.

> ***SQ2:*** *what are the existing quality measures for design rationale?*

The reasoning is a thought process and the Rationale is the product of that process. In this part, we want to know what are the available parameters to measure these outcomes.

> ***SQ3:*** *what is the rationale that designers build during the reasoning process?*

When the parameter to measure Rationale is understood, then the research will focus on the pattern of Design Rationale as the outcome of the design reasoning.

SQ4: How does different design reasoning improve the implicit rationale?

Finally, we want to explore the ways that design reasoning could influence the implicit Design Rationale.

1.2.1 Conceptual Model

The relation between the research question and the concepts related to design discourse is shown in Figure 1.1 (below), the main question (RQ) concerns the link between design reasoning as the process and Design Rationale as the product. The Design Rationale itself could be implicit during the design discourse or presented explicitly in the design documentation. First, the concept from previous research regarding these approaches is gathered and summarized (SQ1), and then the way to measure DR is identified (SQ2). And then the pattern of implicit DR is inspected (SQ3). Finally, we examined these patterns with the process that they are generated (SQ4).

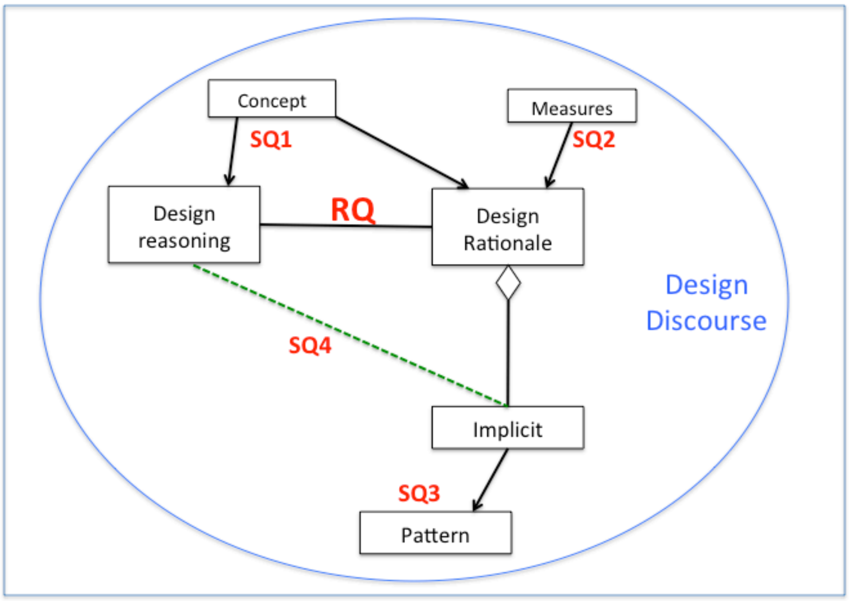


Figure 1.1 Conceptual Model explaining the questions and used concept

1.3 Scientific and Practical Contributions

Practical Contributions The main idea is to encourage designers especially novice designers using these approaches. Implementing the design reasoning techniques especially the problem structuring help designers recognize the issue. Assumption, constraint, risk, and trade-off analysis could be use to increase the arguments that is use to compare two or more design option. Finally, the best option would be the design decision. These reasoning techniques would be a huge addition for novice designers as they also increase their knowledge and perception within design domain.

Scientific Contributions Besides practical contributions, there are some additions to the scientific knowledge base. First of all, the result would expand the knowledge domain of design reasoning and Design Rationale. Furthermore, understanding the link behind the reasoning process and the elements of Design Rationale would give addition to future research focusing on design discourse.

1.4 Thesis Structure

This section has provided the problem statement, research statement and stated the research question of this thesis. The following section will then describe more on the approach and methodology to answer the research questions. In section 3, we provide background literature on Design Rationale and design reasoning while in section 4 provides our data analysis followed by qualitative data analysis and process mining in section 5 and section 6 consecutively.

The Final part of this thesis is started with the result and findings (section 7). This thesis is finished with a conclusion to answer the research questions. The discussion contains limitation and future research directions available in section 8.

2 Research Method

In order to answer the research question, we chose the exploratory experiment as our research method (Franklin, 2005), because we had no preconception as to what hypothesis to test. However, we wanted to discover how the design reasoning affects the Design Rationale elements as the product of the design phase.

2.1 Literature Study

Background knowledge of Design Rationale and their elements is one of the first requirements of this research. The literature study forms the foundation for the whole thesis research. Therefore, a firm and valid method are used for this phase. The process starts by finding related knowledge from significant contributions in the research domain using the most common scientific journal and article databases and search engines (i.e. Google Scholar). The literature study's main focus is on answering SQ1 and SQ2.

The second step in the process continues by finding more literature using the "Snowballing" procedures (Skoglund & Runeson, 2009). It means that to follow the references from or to one paper to find other relevant articles. The snowballing can be forward and backward. Forward means that refers to looking at papers citing the paper that has been found relevant, while backward means that following the reference list of one paper. The information is mainly gathered from several scientific publications available in Section 3.

2.2 Empirical Research- Experiment

The experiments are meant to answer the SQ3 and SQ4, and will ultimately answer the main research question. The experiment design is using guidelines proposed by Wohlin et al. (2012). They mentioned, "In the experiment we could control the situation and manipulate the behavior directly, precisely, and systematically." The experiment involves several steps, such as Scoping, Planning, Operation, Analysis, and Presentation.

2.2.1 Scooping

Our scope of these experiments is in the design reasoning technique and Design Rationale during a design discourse session. With the experiment, we wanted to discover the pattern of Rationale that designers build from reasoning process. Furthermore, we want to know how these design reasoning techniques can influence the implicit and explicit Design Rationale.

2.2.2 Planning

The exploratory experiment was conducted at Utrecht University with Master students during the workshop sessions of Software Architecture course. Within the workshop, students must create and evaluate software architecture. Each session of the experiment was 3 hours. The central content of the experiment is the Irvine experiment that was performed at the University of California, Irvine (UCI, 2010), where several design teams were asked to design a traffic simulator and the process was recorded and transcribed.

In the experiment, there were 12 teams of students that divided into two groups. Most teams hold three students, but some have only two students, and one team has four students. The group selection based on their grades from previous assignment so there is a mix of teams who have shown to do well, and those who have lesser grades.

Before carrying out the experiment, the students had a lecture on Views and Viewpoint (Rozanski & Woods, 2012). To make sure the data is adequate, the experiment split into two sessions: the design session that will result with the implicit Design Rationale, and the evaluate session that explicitly presents the Design rationale.

2.2.3 Operation

First Experiment

The first session was design discourse session, the student given a task to design architecture for traffic simulator software. Although the concepts of traffic lights, lanes, and intersections are common and appear to be simple, building a traffic simulator to represent these relationships and events in real time is complex. The students were allocated randomly to two groups: test group and control group, which in each group will consist of six teams.

Participants in both the test and control group were asked to use a think-aloud method during the design session. Both the control group and the test group designed the Context, the Functional, and the Informational viewpoint of the traffic simulator software within two hours and then they had forty-five minutes to document their design.

During this session, the control group carried the assignment to design the traffic simulator without any design reasoning techniques. The test group gets a design reasoning process using a card game (Schriek, 2016). These cards are meant to prompt the players to question their decisions in order to come to better thoughtful design and also to help in the design discussion. Each team member is going to get a deck of card consists of fifteen cards.

The result of this session is the two hours of recording of design session and documentation of the Context, Functional, and Information viewpoint using the given template (Appendix A). Within the template, we want the students to give an explanation for the design that covers: the model as the representation of the view, the mere description of the view, the glossary of each element in the view, and the Rationale to describe the argument why the view is as it is.

Second Experiment

The second session was to evaluate the design by focusing on the Rationale behind the design. Each team needs to find reasons for the design decisions that they have made during the first assignment (The traffic simulation), and then make their implicit assumptions or rationales into explicit, and then they must clarify their reasoning. The students need to analyze two architectures, one is their own, and one is for another team. They have to find arguments to support and against those different designs. The target of this session is to make argument model that explicitly shows all of the reasoning behind their design decisions, and then the material can be use to compare with the implicit argument from the audio transcript from the first session.

To help visualize the argument, the students get an additional online tool to make the argument diagram. The address of the tool is http://ova.arg-tech.org.

2.2.4 Presentation

When the experiment is completed, the findings will be presented as thesis report. Furthermore, we hope it could be presented as a scientific paper. The structure for the thesis report will consist of: Abstraction as the general description of the research, an Introduction, Methodology, Literature Review, Analysis, Result and Finding, Discussion and Conclusion.

2.3 Validity Issue

Yin (2003) describes four different criteria for empirical research. These criteria are: construct validity, internal and external validity and reliability. This section describes for each of these four validity threats what they imply and the actions performed in this research.

Construct Validity: This aspect refers to the extent in which the used concepts are operationalized and measured correctly. To achieve this, only well-established concepts should be used to construct the theories, or these should be defined sufficiently (Yin, 2003). To secure this validity threat, no new concepts were introduced. The concepts that are used are refined from several literature sources.

Internal validity: This aspect of validity is of concern when causal relations are examined (Yin, 2003). In other words, the data collected is adequate to draw a valid conclusion. We notice the limitation of this experiment include the student's background, student's experience, and their design abilities.

External validity: This aspect of validity is concerned with to what extent it is possible to generalize the findings, and to what extent the findings are of interest to other people outside the investigated case (Yin, 2003). What mitigates this threat is that we made no assumptions about the students' experience or educational background and teams consisted of random mix students. However, we have a low number of participants that could be considered as a threat.

Reliability: This refers to the ability to repeat the research, with the same results (Yin, 2003). This section and the documentation template describe the outline of the experiment. With this, it should be possible to repeat this research in the future.

3 Theoretical Background

This section provides some background on Design Rationale and design reasoning. First, the concept of Design Rationale will be provided in Section 3.1, followed by several previous approaches of capturing Design Rationale. Furthermore, the design reasoning and reflection will be discussed in Section 3.3. Based on this research the framework of Design Rationale elements is developed.

3.1 Design Rationale Concept

The meaning of Rationale from the Cambridge Dictionary is “the reasons or intention that cause a particular set belief or actions”. In other words, we can say that rationale is the justification behind a decision. Dutoit et al. (2007) mentioned that Rationale could serve two different objectives: discourse and knowledge capture. Rationale enables negotiation among stakeholders by showing the possible solution options with their pros and cons. By capturing Rationale, the stakeholders also present the explicit knowledge that usually tacit, which can help people who maintain the system analyze certain decisions and change impacts regarding the system.

In relation to software design, Rationale is captured and used in many forms. The representations of Design Rationale range from informal to formal notations. An informal notation provides data in formats that are easily generated and understand by a human, however, hard to be used by the computer, for example, the natural language. A formal notation allows the computer to use the data but does not always create information in a form that a human can understand. A semi-formal notation attempts to use the advantages of both approaches.

According to Shum (1995, p. 2), Design Rationale (DR) is “a representation of the reasoning behind the design of an artifact”. Conklin and Yakemovic (1991) consider Design Rationale as capturing the history of how a design comes about through recording logical reasoning to support future reference. Carroll and Rosson (1991) suggested that Design Rationale could be viewed as psychological claims that are embodied by an artifact for the situation that it is used. Maclean et.al (1991) claimed that Design Rationale can be a description of the design space and used to deliberate design decisions. Lee (1997) stated that Design Rationale consist of the reasons behind a design decision, the justification of that decision, the other alternatives that have already been considered, the evaluated tradeoffs, and the argumentation that led to the design decision.

Recent research by Tang et al. (2007) also mentioned that Design rationale is the reasoning behind a design, and it provides an explanation of the design. DR shows how the requirements are satisfied, why certain choices of design are selected and how the system architecture influenced by the environmental conditions.

To summarize, most authors in the literature agree that Design Rationale is about the reasoning behind a design (e.g. Shum, 1996; Lee, 1997; Tang, 2007).

3.2 Approach to Capture Design Rationale

The most common approach using Design Rationale is the argumentation-based type (Shum & Hammond, 1994). It uses nodes and links to represent knowledge and relationships. It dates back to Toulmin's model of arguments (Toulmin, 1958) (Figure 3.1), which consists of five components and four relationships. Arguments contain a fact or observation (Datum), which via a logical statement (Warrant), which acts as the "bridge" between datum and claim, allows one to draw a conclusion (Claim). The Warrant supported by The Backing (which indicates the source of the warrants), and the Claim qualified with a Rebuttal (specifying exceptions to the rule).

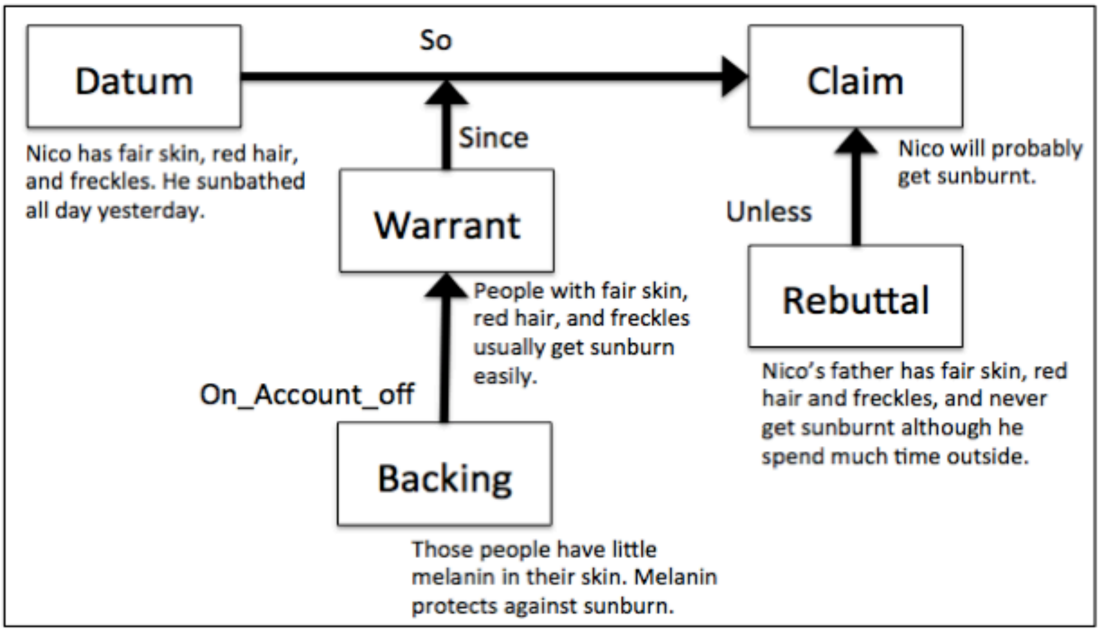


Figure 3.1 Toulmin's model of Arguments (Toulmin, 1958)

Since Toulmin, many similar argumentation-based approaches such as Issue-Based Information System (IBIS), QOC, and DRL have been proposed.

3.2.1 Issue-Based Information System (IBIS)

The Issue-Based Information System (IBIS) proposed by Kunz and Rittle (1970) is a method for structuring and documenting Design Rationale. The concept focuses on solving the issue by cooperation using an argumentative process among the stakeholders. Any problem, concern or question can be an issue that needs discussion in order for a design to continue. This method already implemented and success in several sectors such as architectural design, city planning, and organization planning (Conklin & Begeman, 1988).

The IBIS method structure consists of 3 different node types - Issues, Positions, and Arguments - and eight different link types - Supports, Object-to, replaces, respond-to, generalizes, specializes, questions, and suggested-by. Each issue can have several responding positions that provide a possible answer to the issue, while each position can have one or more arguments to support or to object-to it. Issues can be used to generalize or specialize other issues. Furthermore, issue can also question or suggested-by other issues, positions, and arguments. Figure 3.2 shows a simple example of IBIS method structure.

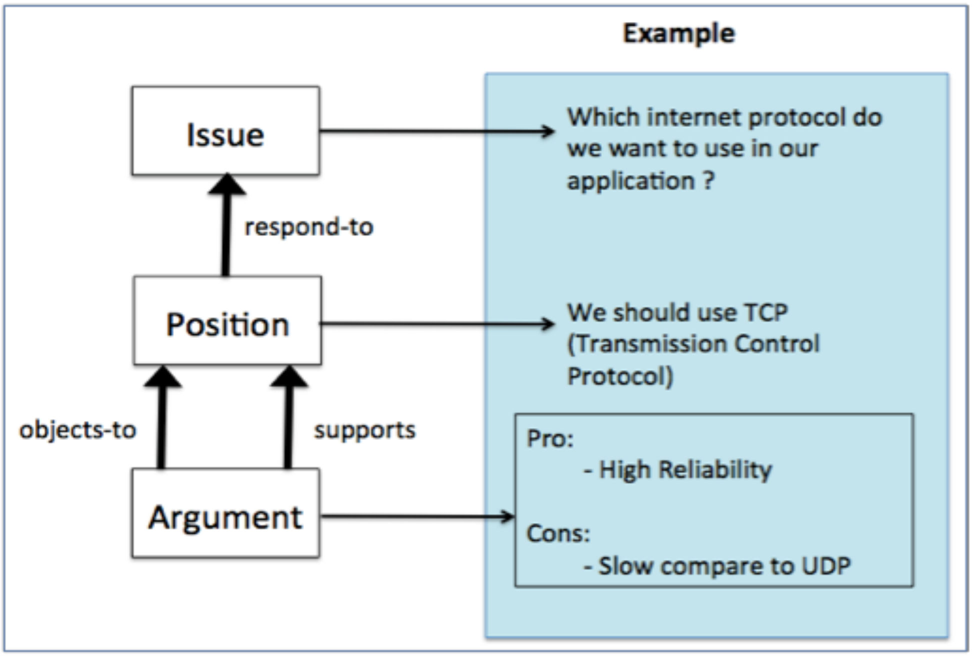


Figure 3.2 IBIS method structure

3.2.2 Question, Options, and Criteria (QOC)

Maclean et al. (1991) present another approach of capturing Design Rationale, using a semiformal notation that contains Questions, Options, and Criteria. They propose the Design Space Analysis (DSA), "analysis that places an artifact in a space of possibilities and seeks to explain why particular artifact was chosen from this possibilities", the QOC notations represent the design space around an artifact. The questions are used to identify the design issue, while the options provide the possible answers to the questions, and the criteria are used to assess and compare the design options.

Figure 3.3 shows an example of QOC (Question, Options, and Criteria). The *Question* is "how wide" the object on the screen, the *Options* is "wide" or "narrow". And the Criteria are "screen compactness" and "ease of hitting with mouse". The positive assessment is whether the Option satisfies the Criteria, while the negative assessment is the opposite. For Example, if the object is wide, it is easy to hit with the mouse pointer, although it will use a lot of space on the screen. If the object is Narrow, it saves space on the screen, but an extra effort will be needed to hit the object with the mouse pointer.

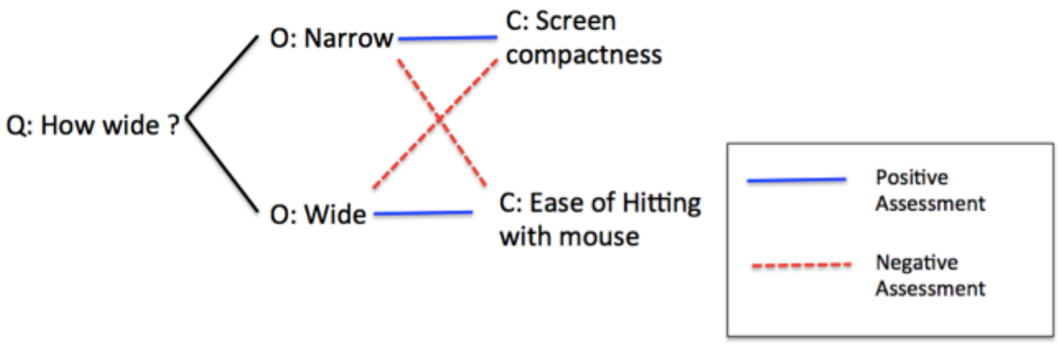


Figure 3.3 QOC Design Rationale example

QOC as an element in Design Space Analysis provides a useful way to organize the information regarding the context of reasoning surrounding a design.

Maclean et al. stated that their approach is different with IBIS method, while IBIS focus in capturing the history of the design deliberations. They approach emphasizes the Design rationale as a knowledge representation of the design space. Conklin and Yakemovic (1991) explain the different as "structure-oriented" and "process-oriented". The QOC focus on Design Rationale as the structure in the design space while the IBIS method focuses on the Design Rationale as a history in the process of the design. They stated that the Design rationale in the structure oriented is prescriptive, in the terms that it summarizes the design decisions and their tradeoffs in order for other to reuse the reasoning. Whereas for the process-oriented described as descriptive in which the reusability is incidental because the Design Rationale itself provided by the unique series of actions during the design.

3.2.3 Decision Representation Language (DRL)

Lee and Lai (1991) proposed DRL as another approach to capture Design Rationale. It is a language to representing and managing the qualitative elements of decision-making. The fundamental objects of DRL are Alternatives, Goals, and Claims. Other Objects in DRL are no less essential for decision-making, but they are special cases, or they are useful beyond the concept of decision-making. For example the Procedure, which represents either an executable procedure or textual description of a procedure. Figure 3.4 shows the complete DRL vocabulary.

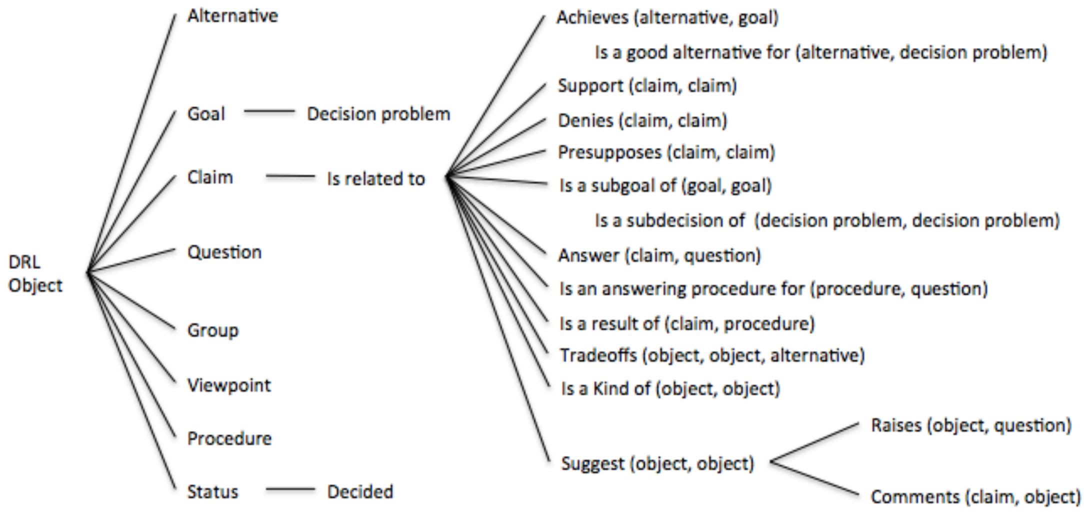


Figure 3.4 DRL complete vocabulary (Lee, 1991)

Alternatives represent the option to choose from, Goals Specify the properties that an ideal option should have, and Claims are used to represent arguments relevant for choosing among the alternatives. Figure 3.5 shows an example of the DRL model.

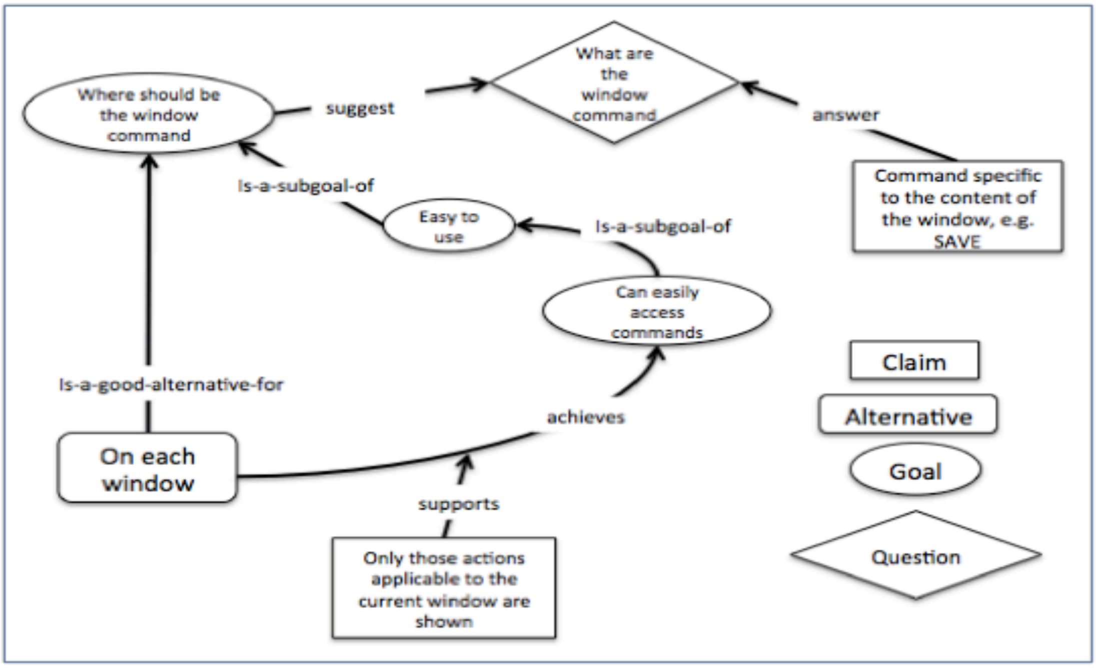


Figure 3.5 Example of DRL model (Adopted from Lee, 1991)

3.2.4 Summary of Previous Approach

From these different argumentation-based Rationale approaches, we try to find the equivalent components to summarize every approach to create the framework of Design Rationale elements. We refine the similar notation as:

- Design Issue, which basically the main target of design deliberation. They represent as Issue in IBIS, as Question in QOC and as Goal in DRL.
- Design Option, the available choice that could be select to accommodate the design target. They represent as Position in IBIS, as Option in QOC and as Alternative in DRL.
- Argument is the value that could assist or impede the design choices. They represent as Argument in IBIS, as Criteria in QOC and as Claim in DRL.

Table 1 Similar component in argumentation-base Design Rationale

	IBIS	QOC	DRL
Design Issue	Issue	Question	Goal
Design Option	Position	Option	Alternative
Argument	Argument	Criteria	Claim

The design phase begins with a set of requirements defining the system being designed. These requirements are then mapped to Design Issue, and then one or more Design Options could satisfy the Design Issue. The Argument for each Design Option is represented as support/positive (pros) or against/negative (cons) for each choice. The best Design Option will become the final Design Decision. Figure 3.6 shows how Design Rationale elements link the requirements and the final Design Decision.

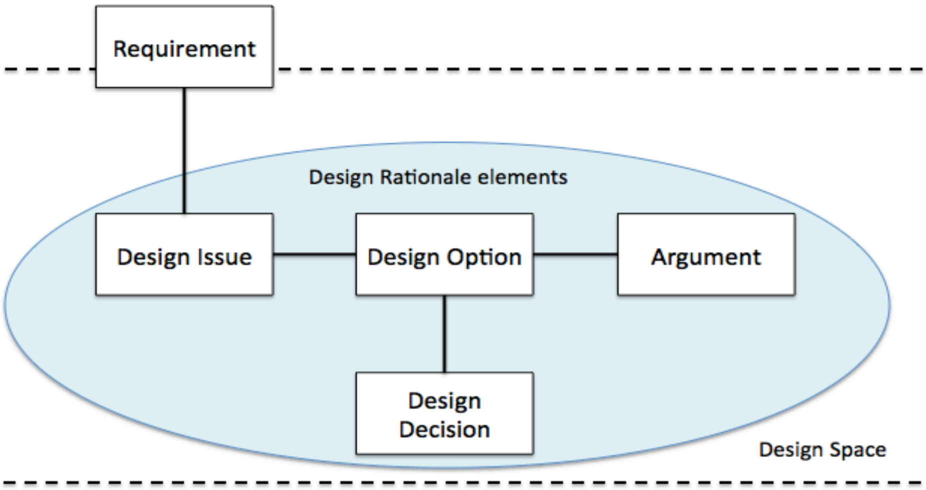


Figure 3.6 The framework of Design Rationale elements

When we trace the similarity with the Toulmin's model of Arguments, the framework of Design Rationale element is focused on the discussion and arguments that happen during the design discourse, and the Toulmin's model originally aimed to develop view of logic behind reasoning that led to graphical format for laying out the structure of an argument.

3.3 Design Reasoning and Reflection

Design reasoning is an important process that designers use in developing a solution (Tang et al., 2008). Designers in the software industry often rely on their experience to make design decisions, if the designers are familiar with the problem then selecting the best design decisions is quite easy. However, if the designers are unfamiliar and new to the problem, then the problem occurs. According to Tang and Van Vliet (2012), if the designers are not familiar with the problem space, then they should start with gather all relevant requirements, then contemplate what problem to solve, then try to create several solution options to address the problem, and then decide which solution option is the best.

Researchers in psychology have proposed that there are two distinct cognitive systems underlying reasoning. The heuristic system (System 1) comprises a set of autonomous subsystems that will tend to solve a problem by relying on prior knowledge and beliefs (De Neys, 2007) while the analytic system (System 2) allows reasoning according to logical standards (De Neys, 2007). Another illustration that we can use on how designers make their design decisions is the reflective thinking (Razavian et al., 2015). They argued that quality of software design depends on the design thinking and the cognitive process of the designers, and they theorize a model of design thinking that consists of two minds of designers. Mind 1 is about logical design reasoning of the designers and Mind 2 is about the ability of reflection the design reasoning itself. Combining the two minds and the design process by Tang and Van Vliet, Razavian et al. (2015) presents the software design-thinking model (Figure 3.7).

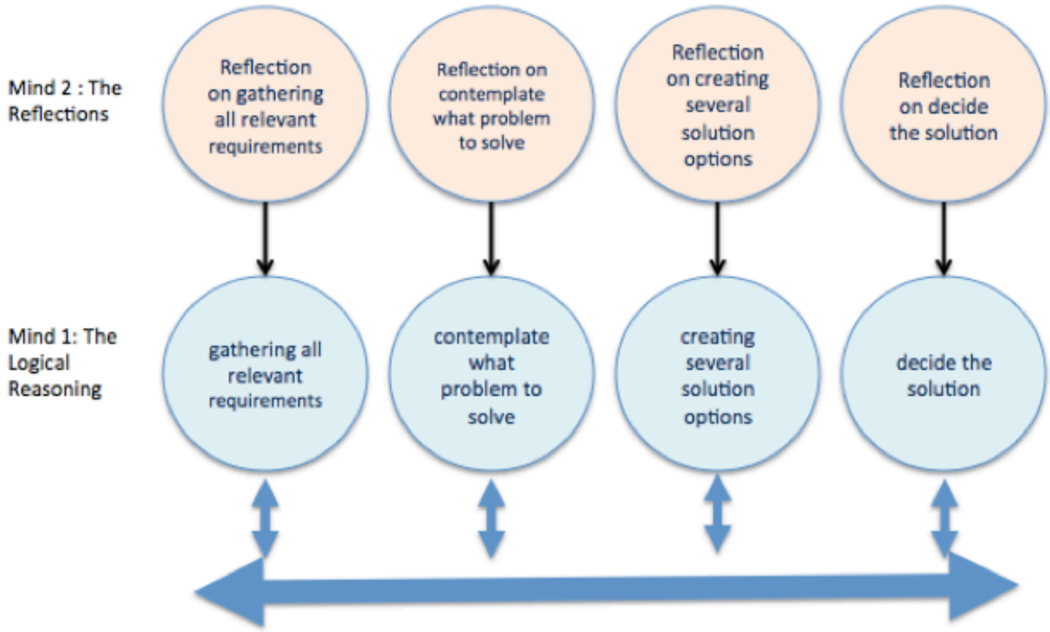


Figure 3.7 The software design-thinking model (Razavian et al., 2015)

A paper by Tang (2011) mentioned several techniques of design reasoning and some of them have already been used in the software industry. These design-reasoning techniques are based on what people think the reasoning issues are, and suggestion on ways to tackle these problems. Those techniques are:

- Assumption analysis. The validity and accuracy of a requirement or a premise are based on whether there is assumptions behind those are clear, and if any hidden assumptions exist it may affect the design. Therefore, it is important for designers to carry out an assumption analysis, which is questioning the possible tacit assumptions that may have been made, consciously or unconsciously.

- Risk analysis. A risk is something that may go wrong, and it can be treated as something that is unknown with a probability that some unwanted conditions can affect a design. However, there are no prescriptions to avoided risk. Therefore, many decisions still have to be made based on some risk mitigation strategies. These strategies are a way for designers to estimate and explicate the effect of risk to the design. A checklist of some significant risks in a software design can be prepared to remind software designers about this risk.
- Constraint analysis. This analysis focuses on the idea that every requirement, system environments, project environments, and organizations may have some constraints on the way that they designed and implemented. These constraints are often tacit and not explicitly discussed or documented during the design. Therefore, it is important for the designers to note these constraints. This note also could be use later to check if there is a conflict in the design.
- Trade-off Analysis, this analysis is required when the designers cannot satisfy all the requirement and constraint at a certain point. Kazman et al. (1998) proposed The Architecture Tradeoff Analysis Method (ATAM) as a method to evaluate the priorities and utility of multiple quality requirements in making a trade-off decision.
- Problem structuring, this technique require the designers to relate the issues and investigate how they influence each other. Understanding the problem, designers minimize the error of overlooking key issue in design. It also could lead designers to find a way to tackle the problem and come out with the solution in the end.
- Option Generation, this technique is required for designers in order to minimize the effect of anchoring, which means that the first impression of a solution that comes to designers mind anchor, and may hard to change and adjust even though it could be inferior. Tang et al. (2008) have found that designers who are prompted to state option made a better design outcome.

According to Tang (2011), the premises or statements that contain in the arguments also influence the reasoning process of decision-making. He also mentioned that there are two types of arguments, deductive and inductive. A deductive argument is "an argument in which the conclusion is claimed to be impossible to be false if the premises are true and the argument is valid" (Tang, 2011). Which mean that the process of reasoning from one or more statements to reach a conclusion, and the conclusion is always true if the argument is valid. For instance, if the premises are: 1. All men are mortal, 2. Socrates is a man. Then the conclusion will be that Socrates is mortal.

Meanwhile, the inductive argument is "an argument in which it is claimed that the conclusion is improbable to be false if the premises are true and the argument is strong" (Tang, 2011). In other words, inductive reasoning is about probability making the conclusion based on generalization from experience. Furthermore, the premises or statements considered as evidence for the truth of the conclusion. Instead of valid or invalid, the inductive arguments are either weak or strong, which describe the probability of the conclusion to be true. As an example, if the premises are: 1. We have use 20 of X printer for two years, 2. All of them work perfectly, 3. We will add another of X printer in our office. So probably the conclusion will be: This new X printer will be working perfectly for the next two years.

3.4 Connection Between Design Reasoning and Design Rationale

Connecting the concept of design reasoning with Design Rationale, we could say that the design reasoning is the process that helps the designer to make better design decisions while the Design Rationale is the justification, the other alternatives, and the argumentation that leads to the design decision.

For example, when designers compare the design options to select the standalone architecture or the client-server architecture, they can assume that the standalone applications would be cheaper to create however hard to maintain in the future, while building the client-server applications may cost a lot more and require more resources, the maintainability of this architecture is more convenient than the standalone. The trade-off analysis can help the designers to choose between the cost and maintainability. However, whatever the final decision that designers select, the arguments developed when the selection of this design options present their pros and cons. The design issues, the design options, the pros, the cons contribute as the rationale behind the final designer's decision.

3.5 Context and Requirement Quality

The reasoning is a thought process. The results of such thought process in the design phase are design decisions, and these decisions could be good or even bad. Paul et al. (2006) argued that a bad design decision happens partially due to poor quality contexts and requirements. Therefore, they should meet some quality criteria. These criteria are:

- Accuracy. This means that contexts and requirements should free from errors or distortions, in other words, it should be true. The designers should make sure that every context or requirement are genuine and accurate, and not base on some personal belief.
- Relevance. This means that the contexts or requirements are related to the matter at hand.
- Adequacy. This means that the contexts or requirements should complete, any missing one could lead to incorrect conclusions or decisions.

When the contexts or requirements lack these qualities, the design decisions that are based on them can be faulty and defective. However, the problem here is that these measurements only target the context and the requirement. Therefore, we still need to find an appropriate way to measure and quantify the elements of Design Rationale as a whole.

4 Data Analysis

As described in section 2, experiments were performed to help us answer the research questions. The first experiment has resulted in 12 recording of design discourse session and transcribed in nearly 600 pages of transcript, while the second experiment produced 12 arguments diagram, the analysis described in this section are based on this dataset. This dataset is separately attached to the final thesis document. Table 2 provides the detailed description of the transcripts information.

Table 2 Descriptive of the 1st experiment transcripts

Group	Duration of recording	Pages	Word count
C1	1:43:52	26	6874
C2	1:57:15	27	8426
C3	1:22:49	27	10200
C4	1:30:36	48	11099
C5	1:17:20	57	10470
C6	2:06:02	56	14160
T1	2:01:17	93	20306
T2	2:43:00	80	17103
T3	2:01:43	71	17564
T4	3:47:46	50	16067
T5	1:54:48	30	10856
T6	1:51:34	43	13312

To investigate the link between design reasoning and Design Rationale from the first experiment, we analyzed each team transcript. The method that we chose for this part is transcript coding method of Miles and Huberman (1994), which they stated as *"to review a set of field notes, transcribed and to dissect them meaningfully"*.

Next section provides the codes that we use, and then followed with the coding procedure in order to carry out the coding systematically. Section 4.3 presents the result of the coding consistency. Section 4.4 displays the coding result, followed by the quantitative data analysis in Section 4.5. An example of graph plotted from the result of the codes presented in Section 4.6. Finally, the conclusion of this section presented in Section 4.7.

4.1 Codes

According to Miles and Huberman (1994), codes are *"tag or labels for assigning unit of meaning into descriptive or inferential information compiled during a study"*. For creating the codes, we followed their suggestion, which is creating a provisional ‘start-list’ from concept and variable that we have from the study and then it is refined during the analysis. By coding the transcripts, we mapped each of design discourse sessions on the design reasoning technique and the framework of Design Rationale elements.

The codes consist of two major categories: design reasoning techniques and Design Rationale elements. The design reasoning techniques consist of the techniques presented in the section 3.3, which are: (i) Problem Structuring, (ii) Option Generation, (iii) Trade-off Analysis, (iv) Assumption Analysis, (v) Constraint Analysis, and (vi) Risk analysis.

While the Design Rationale element base on the framework of design rationale presented in section 3.2.4, which consists of: Design Issue, Design Option, Design Decision and Argument. Coding the Argument, we directly translate it into the actual product of each analysis, they are: Assumption form the Assumption Analysis, Constraint from the Constraint Analysis, Risk form the Risk Analysis, and Pro and Con from Trade-off Analysis. Table 3 shows the category of codes that consist of Design reasoning techniques and Design Rationale elements.

Table 3 List of Codes

Category	Code	Name	Description
Design reasoning techniques	PS	Problem Structuring	Identifying the key issue of design
	OG	Option Generation	Discussing the options available for design solutions
	TA	Trade-off Analysis	Weighing the pros and cons concerning the design to come to a decision
	AA	Assumption Analysis	Questioning the premises of the requirements and context, the validity of arguments
	CA	Constraint Analysis	Identifying constraints in the design and how these constraints influence the design
	RA	Risk Analysis	Identifying risks in the design and how to mitigate those risks
Design Rationale elements	DI	Design Issue	A design issue
	DO	Design Option	An option for solution
	DD	Design Decision	A design decision
	A	Assumption	A supposition that is taken for granted or questioned to
	C	Constraint	A restriction on the condition of the design

	R	Risk	An aspect of the design which is identified to be problematic
	PRO	Pro	A design support for a proposition
	CON	Con	A design support against a proposition

4.2 Coding Procedure

Two researchers independently coded each transcript. To make coding systematic, we used the following coding procedure:

1. Step 1: Transcribe the audio recording material. We transcribe the discussion session and the time of each team audio file.
2. Step 2: Coding the design reasoning techniques. This stage codifies the design reasoning techniques based on students design discussion and their time stamp.
3. Step 3: Coding the Design Rationale elements. This step encodes the Design Rationale elements and the time that it's occurred, observed from students design discussion.

The transcripts will independently coded by two researchers and then we will check the coding consistency using inter-coder agreement (Cohen, 1968).

4.3 Coding Consistency

To ensure that the interpretation of the design reasoning and Design Rationale elements were consistent among the coders we conducted the inter coder reliability. After the two researchers separately finished encoding each transcript, we checked the coding consistency of the design reasoning and Design Rationale elements using the Cohen's Kappa (Cohen, 1968). The average result of coefficient was 0.64 (Table 4), which indicate substantial agreement between the two researchers. This measure provided assurance that the interpretations of design reasoning and Design Rationale elements were consistent among the coders.

Table 4 Inter coder Reliability result

Group	Kappa	Agreement (%)	A and B (%)	Not A and Not B (%)	Disagreement (%)	A and Not B (%)	B and Not A (%)
Group 1	0.60	97.33	2.52	94.81	2.67	1.77	0.90
Group 2	0.76	98.15	2.81	95.35	1.85	0.83	1.02
Group 3	0.66	98.39	2.23	96.16	1.61	0.61	1.00
Group 4	0.65	97.31	2.84	94.47	2.69	1.54	1.15
Group 5	0.67	96.89	2.54	94.35	3.11	1.06	2.05
Group 6	0.67	97.68	3.10	94.58	2.32	0.65	1.67
Group 7	0.61	97.35	1.86	95.50	2.65	0.98	1.67
Group 8	0.60	97.13	2.80	94.33	2.87	1.53	1.34
Group 9	0.61	97.08	2.46	94.62	2.92	1.02	1.90
Group 10	0.62	95.54	2.77	92.78	4.46	2.63	1.82
Group 11	0.63	96.01	3.24	92.76	3.99	1.82	2.18
Group 12	0.61	97.35	2.52	94.83	2.65	1.57	1.08
Average	0.64	97.18	2.64	94.54	2.82	1.33	1.48

4.4 Coding Result

After finished with the coding consistency between the two researchers, the result of the codes can be presented as two separate tables, the reasoning technique summary (Table 5) and the Design Reasoning elements summary (Table 6). Furthermore, this result can be used for a quantitative measurement in the next section.

Table 5. Coding result of design reasoning techniques

Reasoning Technique	Control Group						**Total**	**Average**	Test Group						**Total**	**Average**
	C1	C2	C3	C4	C5	C6			T1	T2	T3	T4	T5	T6		
Assumption Analysis	2	0	2	2	1	2	9	1.50	8	4	6	4	6	5	33	5.50
Constraint Analysis	3	4	9	10	6	3	35	5.83	5	6	1	6	7	7	32	5.33
Option Generation	1	6	2	8	7	6	30	5	10	2	7	8	7	8	42	7
Problem Structuring	12	27	31	12	21	17	120	20	24	23	26	28	22	26	149	24.83
Risk Analysis	2	4	3	3	2	3	17	2.83	5	7	2	8	6	6	34	5.67
Trade-off Anaysis	1	3	0	5	0	0	9	1.5	3	0	1	4	3	0	11	1.83
Total	21	44	47	40	37	31	220	36.67	55	42	43	58	51	52	301	50.17
Average	3.50	7.33	7.83	6.67	6.17	5.17	36.67	6.11	9.17	7.00	7.17	9.67	8.50	8.67	50.17	8.36

Table 6. Coding result of Design Rationale elements

DR Element	Control Group						**Total**	**Average**	Test Group						**Total**	**Average**
	C1	C2	C3	C4	C5	C6			T1	T2	T3	T4	T5	T6		
Assumption	2	0	2	6	2	2	14	2.33	8	5	7	5	9	4	38	6.33
Con	2	3	0	11	2	4	22	3.67	11	1	5	10	3	1	31	5.17
Constraint	3	9	14	13	20	15	74	12.33	6	13	4	10	13	11	57	9.50
Design Decision	4	8	16	21	16	8	73	12.17	31	9	14	17	8	13	92	15.33
Design Issue	4	8	16	21	16	8	73	12.17	31	9	14	17	8	13	92	15.33
Design Option	7	10	19	22	22	17	97	16.17	40	13	20	24	14	25	136	22.67
Pro	7	7	9	16	2	3	44	7.33	22	7	11	13	5	3	61	10.17
Risk	2	3	4	4	2	7	22	3.67	6	8	7	8	5	7	41	6.83
Total	31	48	80	114	82	64	419	69.83	155	65	82	104	65	77	548	91.33
Average	5.17	8.00	13.33	19.00	13.67	10.67	69.83	11.64	25.83	10.83	13.67	17.33	10.83	12.83	91.33	15.22

4.5 Quantitative Data Analysis

To get more insight with the data, we conduct a quantitative data analysis using a 2-sample t-test. It is applied to compare whether the mean difference between two groups is really significant or if it is due instead to random chance. In other words, we want to know the effect of treatment for each design reasoning techniques and Design Rationale elements.

Using the data from Table 5 and Table 6 to determine whether the means of two groups are different, we compare every design reasoning techniques and Design Rationale elements (the detail of every t-test available in Appendix E). As an example we show the step for compare the means of the Assumption from the Design Rationale elements, the step that we conduct as follows:

1. The first step we state the null and the alternative hypothesis.

   Ho: $\mu 1 = \mu 2$, (there are no difference between the means)

   Ha: $\mu 1 \neq \mu 2$, (the population means is different)

   For this analysis, the significance level is 0.05, and the test method is 2-sample t-test.

2. The data that we use
   Assumption of Control Group:

   Data: 2, 0, 2, 6, 2, and 2
   Mean: 2.33 Standard deviation: 1.97

   Assumption of Test Group:

   Data: 8, 5, 7, 5, 9, and 4
   Mean: 6.33 Standard deviation: 1.97

3. Calculation

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	2.3333333	6.3333333
Variance	3.8666667	3.8666667
Observations	6	6
Pooled Variance	3.8666667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-3.523321	
P(T<=t) one-tail	0.0027539	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.0055078	
t Critical two-tail	2.2281388	

4. Result of the calculation shown that P-value was 0.005, and it is under the significant level of 0.05. Therefore, we reject Ho and accept Ha, that the mean of populations is different.
5. Conclusion, After conducting 2-sample t-test, we found that there was a significant difference in the number of Assumption made by the control group (M = 2.33, SD = 1.97) and the test group (M = 6.33, SD = 1.97); $t(10) = -3.523$, p= .005. These results suggest that the treatment does have an effect on number of Assumption made by designers. The treatment helps designers create more Assumptions during the design discourse session.

After conducting 2-sample t-test for every design reasoning techniques and Design Rationale elements, we summarize the result in Table 7 and Table 8. The summary presents the data, the mean, the standard deviation (STD), and the P-value of every design reasoning technique and Design Rationale elements that were used for the calculation. As a reminder of P-value, if the P-value were 0.0102, that indicates the difference observed would only be seen about 1.02 % of the time. Given that is a pretty low percentage, researchers conclude that the difference observed is not due to chance and call it statistically significant.

Table 7. t-test result of design reasoning techniques

Reasoning Technique	Control Group							Test Group							P value
Assumption Analysis	Data :	2	0	2	2	1	2	Data:	8	4	6	4	6	5	0.0002
	Mean:	1.5		STD:		0.84		Mean:	5.5		STD:		1.52		
Constraint Analysis	Data :	3	4	9	10	6	3	Data:	5	6	1	6	7	7	0.7538
	Mean:	5.83		STD:		3.06		Mean:	5.33		STD:		2.25		
Option Generation	Data :	1	6	2	8	7	6	Data:	10	2	7	8	7	8	0.2375
	Mean:	5		STD:		2.83		Mean:	7		STD:		2.68		
Problem Structuring	Data :	12	27	31	12	21	17	Data:	24	23	26	28	22	26	0.1774
	Mean:	20		STD:		7.85		Mean:	24.83		STD:		2.23		
Risk Analysis	Data :	2	4	3	3	2	3	Data:	5	7	2	8	6	6	0.0102
	Mean:	2.83		STD:		0.75		Mean:	5.67		STD:		2.07		
Trade-off Anaysis	Data :	1	3	0	5	0	0	Data:	3	0	1	4	3	0	0.7682
	Mean:	1.5		STD:		2.07		Mean:	1.83		STD:		1.72		

The explanation of Table 7 (above) is that, the treatment shown an effect in Assumption Analysis and Risk Analysis that designers made, because their P-Value is under the significant level (0.05). However, other techniques do not affected by the treatment. Therefore, we can conclude that the treatment has an effect only on the Assumption Analysis and Risk Analysis made by designers. In other words, the treatment helps designers conduct more of Assumption Analysis and Risk Analysis during the design discourse session.

Table 8. t-test result of Design Rationale elements

Design Rationale Element	Control Group							Test Group							P value
Assumption	Data :	2	0	2	6	2	2	Data:	8	5	7	5	9	4	0.0055078
	Mean:	2.33		STD:		1.97		Mean:	6.33		STD:		1.97		
Con	Data :	2	3	0	11	2	4	Data:	11	1	5	10	3	1	0.5429357
	Mean:	3.67		STD:		3.83		Mean:	5.17		STD:		4.40		
Constraint	Data :	3	9	14	13	20	15	Data:	6	13	4	10	13	11	0.337035
	Mean:	12.33		STD:		5.79		Mean:	9.5		STD:		3.73		
Design Decision	Data :	4	8	16	21	16	8	Data:	31	9	14	17	8	13	0.4797209
	Mean:	12.17		STD:		6.46		Mean:	15.33		STD:		8.36		
Design Issue	Data :	4	8	16	21	16	8	Data:	31	9	14	17	8	13	0.4797209
	Mean:	12.17		STD:		6.46		Mean:	15.33		STD:		8.36		
Design Option	Data :	7	10	19	22	22	17	Data:	40	13	20	24	14	25	0.2027519
	Mean:	16.17		STD:		6.31		Mean:	22.67		STD:		9.83		
Pro	Data :	7	7	9	16	2	3	Data:	22	7	11	13	5	3	0.4338005
	Mean:	7.33		STD:		5.01		Mean:	10.17		STD:		6.88		
Risk	Data :	2	3	4	4	2	7	Data:	6	8	7	8	5	7	0.0054631
	Mean:	3.67		STD:		1.86		Mean:	6.83		STD:		1.17		

An explanation of Table 8 (above) is that the treatment showed an effect in Assumption and Risk that designers made, because their P-Value is under the significant level (0.05). Therefore, the treatment has an effect on a number of Assumption and Risk made by designers. The treatment helps designers create more Assumptions and Risks during the design discourse session. However, other elements seem do not affected by the treatment because their P-value was above the significant level.

4.6 Data Display

We plotted the design reasoning and Design Rationale elements over the observed period of about two hours each session. The resulting graphs (Complete graph shown in Appendix B) portray the design reasoning and Design Rationale elements over the design discourse. The y-axis shows the design reasoning or the Design Rationale elements, while the x-axis presents the time. For example, the design reasoning of transcript 12 is visualized in Figure 4.1 and Design Rationale elements of transcript 12 presented in Figure 4.2.

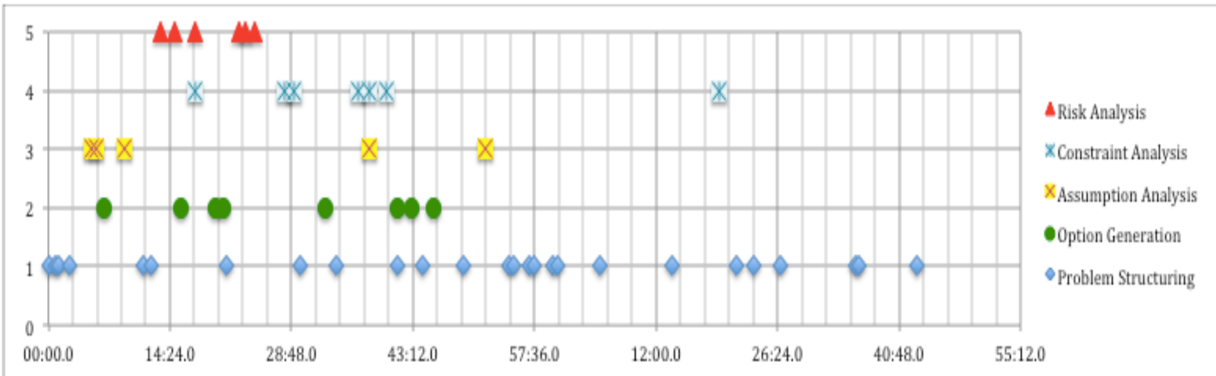


Figure 4.1 Design reasoning Time plot of transcript 12

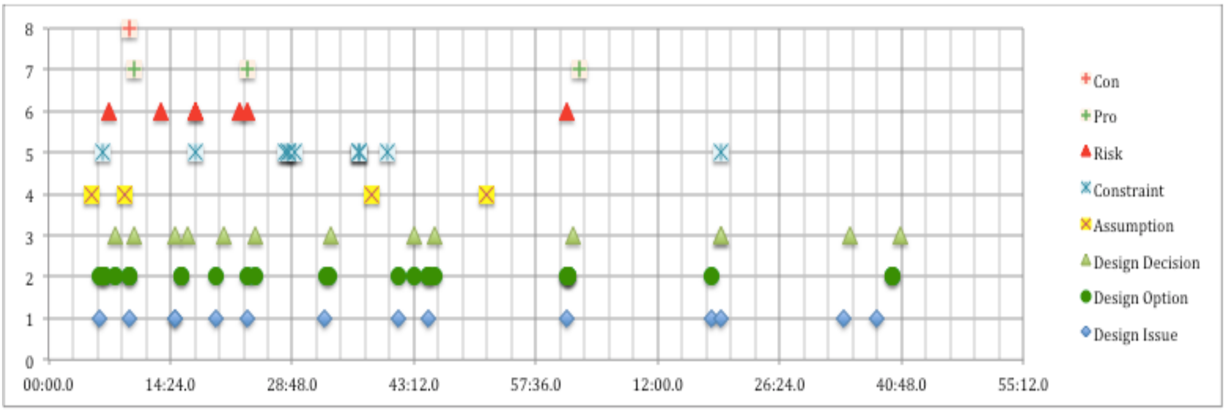


Figure 4.2 Design Rationale elements time plot of transcript 12

4.7 Conclusion

This section has treated the data from the result of the first experiment. The observation is that the treatment with the reflective method of design reasoning has an effect on some techniques. The affected techniques are Assumption Analysis and Risk Analysis. Meanwhile, the elements of Design Rationale that affected by the treatment are Assumption and Risk.

5 Qualitative Data Analysis

In this thesis, we also use the qualitative data analysis method of Miles and Huberman (1994), they mentioned that the qualitative data analysis has the advantage to show *"what led to what"*, or in other words it can preserve the chronological flow and help the researcher to see which events led to which consequences. Furthermore, we were interested to know what is the rationale that designers build during the reasoning process (SQ3), and how these elements are connected and how they influence each other. Therefore, we want to link these events that happened in the design discourse session and then identify the pattern that appears from it.

5.1 Event-listing Matrix

One of the methods to display the data from the qualitative analysis of Miles and Huberman (1994) is the event-listing matrix. This matrix focuses on understanding a chronology. Event-listing matrix arranges a series of events sorting them into several categories and help to understand the flow and the connection of events.

We use this matrix to display a set of the events (the elements of Design Rationale) from each transcript, and it helps us to interpret what happened during the design discourse. Designing the matrix, we put a set of event happened during the design discourse session for the rows of the matrix. And for the columns, we incorporate "Design Issue", Design Option", Argument" with their effect (Positive or Negative), and "Design Decision" as the categories.

Entering the data for the matrix we extracted the codes from the transcript and sorted them by the time of appearance. We filled the matrix for each group base on the codes from the transcript and the categories that we already design. With the presence of time for each code from the transcript, we can relate every Design Issue to the Design Decision and Design Option. We also can trace the Argument for each Design Option if they exist. However, after inserting some codes into the matrix there are several Arguments that do not connect to any Design Option. For Example the event no 1 of transcript 12 (Table 9), they made an "Assumption" of the application for use in Netherlands. However, after examining the requirement they found out that the software is for a course in UCI, USA. So this Assumption that they made before become invalid and did not connect to any Design Option.

Table 9 shows an example of this matrix for transcript 12 (the complete matrix of each team available in Appendix C).

Table 9 Example of the event-listing matrix of transcript 12

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			we need to make an assumption that is in the Netherlands, because traffic always every where different	Assumption	None	
2	I'm not sure how	country dependent.	also a constraint because your limit your self to the Dutch driving laws.	Constraint	Negative	being a primarily for a course at UCI, in the first version I think is efficient to make it specific for the united states .
			constraint is region country	Constraint	Positive	
		you can either chose to make it generic and make the rule set for region pop able so you can do region for Germany and for Holland and for many other regions.	so the trade off being that make it modulear will be more expensive	Risk	Negative	
		you can chose to just limit your self to the Netherlands and that's it.				
		it's the university of California Irvine. so I think we should deliver our self in to the united states then.				
3			user language is also the same like for the traffic rules it's will be American English.	Assumption	None	
4	how the application will be deployed	desktop application	old fashion and very problematic if some people have mac and some people have windows. then you have to go through all problem do java application.	Con	Negative	so I think we should make web application.
		web application	because in app you have same problem with different platforms	Pro	Positive	
			if we implement the wrong business rules it will be avoid a traffic response for how the traffic rules are we are and then the whole application is useless and this is I think the biggest risk.	Risk	Negative	
			the business rules being wrong.	Risk	Negative	
			wrong implement and then the system was useless	Risk	Negative	
5	what do you do we want like a lecturer can change this or the developers can change who should be the authority and the context that can change this.	Ideally you make the system somewhere that professor can make rules himself that I think it might be very				I think we should make those rules very modulear so you can easily swap without other rules and that there not entangle in the rest of system so if you make a mistake you can easily fix it. okay. so solution for this problem would be to make the traffic rules dynamic and editable by end user being the professor.
		so we should look into that if that possible if you can make the rules set dynamic and so professor can make it at his own rules and constraints to the traffic light simulator.				

5.2 Pattern Classification

Creating patterns of rationale that designer made during the design discourse session, we analyze the event-listing matrix of each group. To help us identify the pattern that appears, we create some steps and parameters to classify every event. The step that we conducted as follows:

1. First, we split the events on parameter whether they "contain Design Issue" or not (P1). If yes then it went to the next parameter, if not then it is classified as Type- E.
2. Second, the event that has Design Issue as the result from the first step then split again by parameter do they "contain Design Option" or not (P2). If yes then it went to the next parameter, if not then it is classified as Type- A.
3. The third step, the event that has Design option then split again by parameter "contain Design Option more than 1" or not (P3). If yes then it went to the next parameter, if not then it is classified as Type- B.
4. Finally, every event that has Design Option more than 1, we split again with parameter whether "every Design Option contain Argument" or not (P4). If yes then it is classified as Type-D, if not then it is classified as Type- C. Explanation of each type presented in Section 0.

Moving row to row, we analyze every event and classify them. Figure 5.1 provides a visual in form of a classification tree for the parameters and the steps on how we plot each event.

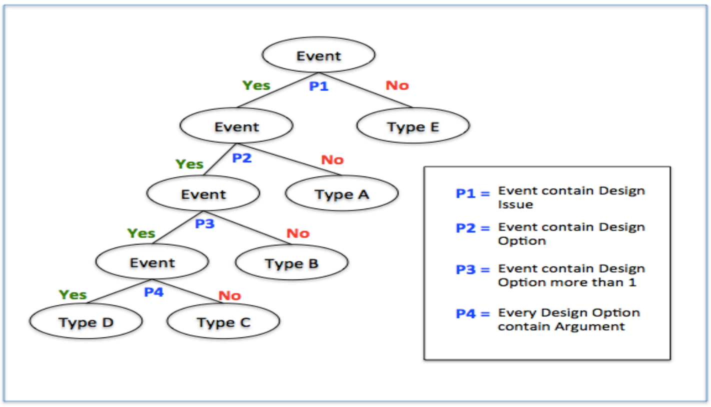


Figure 5.1 Pattern Classification tree

5.3 Pattern Type

This section presents the explanations of pattern types that we discover from the event-listing matrix, we also give the representation of the model for each type with the relationship cardinality of the Design Rationale elements.

5.3.1 Type A

This type presents the situation when designer made the Design Decision directly without any Design Option. We name this pattern "Direct Decision".

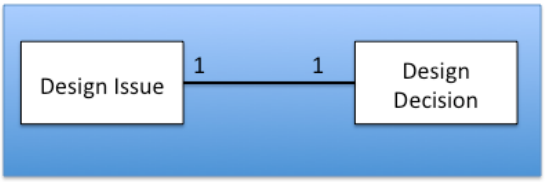


Figure 5.2 Pattern Type A

The example of this type from the event-listing matrix is as follow:

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
15	But we don't know how we're going to visualize the architecture which Is our main concern at this point.					I think we should have a module like the builder module, that allows you to place roads, intersections

5.3.2 Type B

This type shows the situation that Design Decision considered after one Design Option has been proposed. This Design Option may have Argument behind it. We called this pattern as "Focus Decision".

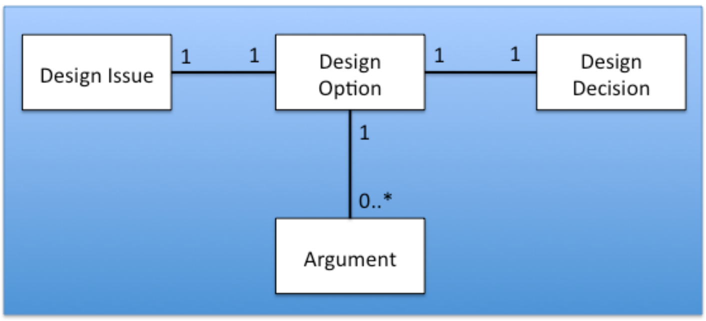


Figure 5.3 Pattern Type B

The example of this type from the event-listing matrix is as follow:

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
9	Can you share your map	No, don't make it too difficult	Out of scope	Pro	Positive	So you only can save and open a file

5.3.3 Type C

This type shows the situation that Design Decision considered after two or more Design Option have been proposed, however, one or more Design Option does not have any Argument that connects to it. We called this pattern "Half consideration".

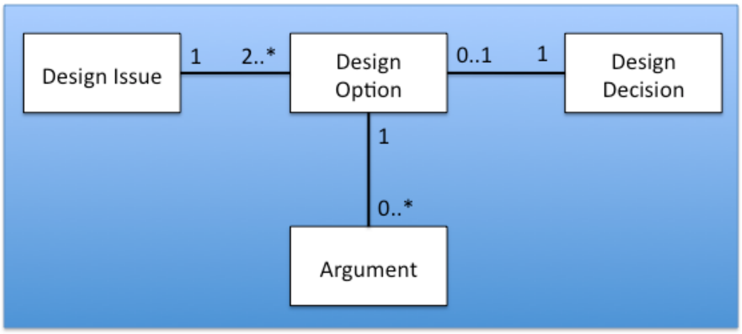


Figure 5.4 Pattern Type C

The example of this type from the event-listing matrix is as follow:

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
14	how are you gonna see this	traffic view				traffic timing scheme
		traffic simulation view				
		traffic timing scheme				

5.3.4 Type D

This type presents the full coverage, which means that the Design Decision taken after more than one Design option was considered and each Design Option has their Design Support. We called this pattern as "Full Consideration".

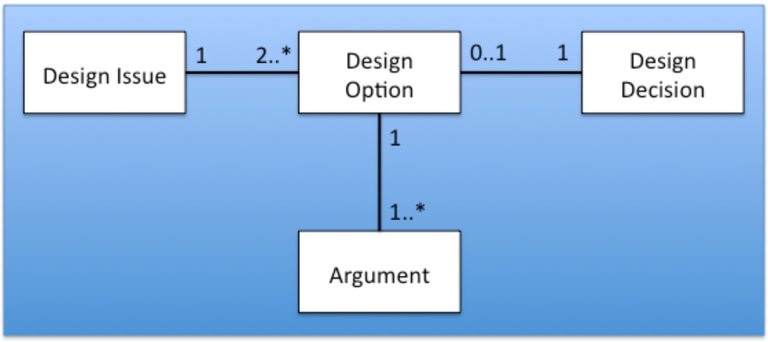


Figure 5.5 Pattern Type D

The example of this type from the event-listing matrix is as follow:

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
4	how the application will be deployed	desktop application	old fashion and very problematic if some people have mac and some people have windows. then you have to go through all problem do java application.	Con	Negative	so I think we should make web application.
		web application	because in app you have same problem with different platforms	Pro	Positive	
			if we implement the wrong business rules it will be avoid a traffic response for how the traffic rules are we are and then the whole application is useless and this is I think the biggest risk.	Risk	Negative	
			the business rules being wrong.	Risk	Negative	
			wrong implement and then the system was useless	Risk	Negative	

5.3.5 Type E

This type shows the Argument that is not directly connected with any Design Issue, Design Option, or Design Decision. We name this pattern as "No link". Table 10 presents the summary of the major pattern discovered.

Table 10 Summary of Pattern

Type	Explanation	Control Group						Total	Average	Test Group						Total	Average
		C1	C2	C3	C4	C5	C6			T1	T2	T3	T4	T5	T6		
A	Direct Decision (Design Issue - Design Decision)	0	2	3	7	1	1	14	2.33	7	2	2	3	1	2	17	2.83
B	Focus Decision (only one Design Option with or without support)	2	2	7	7	9	1	28	4.67	7	4	4	9	1	2	27	4.50
C	Half Consideration (with more than one Design option, however not all have support)	1	1	4	3	6	2	17	2.83	11	2	6	3	5	7	34	5.67
D	Full Consideration (with Design option with support for each)	1	3	2	4	0	4	14	2.33	6	1	2	2	1	2	14	2.33
E	No link (support not connected to anything)	3	0	7	2	4	14	30	5.00	5	2	5	7	10	4	33	5.50
Total		7	8	23	23	20	22			36	11	19	24	18	17		

5.4 Detail Pattern Variant

Analyzing the patterns in more detail, we conduct further examination on the event-listing matrix. Our aim is to identify the variant that appears from each type. The step that we take as follows:

1. First, we make a list of every event with categories that we want to explore, which is the Pattern Type, the number of Design Option, and the effect of Argument for Design Option.
2. Converting the effect of Argument of each Design Option into notation, we convert them using the combination of “number” and ”symbol”. For example, the event number 2 from Table 9 has four Design Options; the first Design option has the argument with one positive and one negative, we plot this as “1+1-“. The second Design Option has one negative Argument, we plot this as “1-“, while the Design Option three and four do not have any Argument, this we plot as “0” for each. So the combination of Argument pattern of event 2 is “1+1-, 1-, 0, 0”. As an example of conversion from Table 9 are presented below (Table 11).

Table 11 Example of plotting Argument Pattern

Team	Event number	Type	DO Quantity	Argument Pattern
Team 12	1	E	None	Assumption
Team 12	2	C	4 DO	1+1-, 1-, 0, 0
Team 12	3	E	None	Assumption
Team 12	4	C	2 DO	1+3-, 1-
Team 12	5	C	2 DO	None

3. We combine the result of conversion of every team and identify the result.

Table 12 shows the number of Design Option for each Major Type from the two groups.

Table 12 List of Type with the quantity of Design Option

Type	Number of Design Option Control Group					Grand Total
	1 DO	2 DO	3 DO	4 DO	None	
A					14	14
B	28					28
C		14	2	1		17
D		13	1			14
E					30	30
Grand Total	28	27	3	1	44	103

Type	Number of Design Option Test Group						Grand Total
	1 DO	2 DO	3 DO	4 DO	5 DO	None	
A						17	17
B	27						27
C		28	3	2	1		34
D		11	2	1			14
E						33	33
Grand Total	**27**	**39**	**5**	**3**	**1**	**50**	**125**

While Table 13 and

Table 14 present the argument pattern in detail for each group.

Table 13 Argument pattern of control group

Type	Argument Pattren of Control Group																														Total
	1-	2-	3-	1+	2+	3+	1+1-	1+2-	2+2-	3+2-	1-, 0	1+, 0	2+, 0	8+, 0	1+, 1-	1+, 2-	3+, 2-	1+, 1+	1+1-, 0	1+1-, 1-	1+2-, 1-	2+1-, 1-	2+1-, 1+	1+1-, 1+	1+2-, 0, 0	3+2-, 2+1-, 1+	2-, 0, 0, 0	Assumption	Constraint	Risk	None	
A																															14	14
B	1	1	1	7	2	1	3	1	1	1																					9	28
C											2	1	1	1					1						1		1				9	17
D															2	2	1	2		1	1	1	1	2		1						14
E																												5	23	2		30
Grand Total	1	1	1	7	2	1	3	1	1	1	2	1	1	1	2	2	1	2	1	1	1	1	1	2	1	1	1	5	23	2	32	103

Table 14 Argument pattern of test group

Type	Argument Pattern of Test Group																																					Total
	3-	1+	3+	6+	1+1-	1+2-	2+1-	9+3-	10+8-	1-, 0	2-, 0	1+, 0	2+, 0	3+, 0	1+, 1-	2+, 1-	2+, 2-	1+, 1+	2+, 1+	2+1-, 0	2+3-, 0	4+2-, 0	1+1-, 1-	1+2-, 1-	1+3-, 1-	2+2-, 2-	1+, 1-, 0	1+, 1-, 1-	1+, 1+, 1-	1+1-, 1-, 0	2-, 0, 0, 0	1+1-, 1-, 0, 0	4+1-, 3+2-, 3-, 2-	2+, 1-, 1-, 0, 0	Assumption	Constraint	Risk	None	
A																																						17	17
B	1	7	1	1	1	1	1	1	1																													12	27
C										1	2	3	2	1						1	1	1					1			1	1	1		1				17	34
D															2	2	1	1	1				1	1	1	1		1	1				1						14
E																																			14	16	3		33
Grand Total	**1**	**7**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**3**	**2**	**1**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**14**	**16**	**3**	**46**	**125**

Simplification of Table 13 and

Table 14 presented below. The simplification made by combining the same argument pattern and ignores the numbers, for example, we combine the frequency of 3+ (three positive) and 6+ (six positive) into the frequency of + (positive). Another example, we combine the frequency of 1+1-, 1- (one positive one negative and one negative) with the frequency of 1+2-, 1- (one positive two negative and one negative) which result in the frequency of +-, - (positive negative and negative).

Table 15. Simplification of pattern variant of control group

Type	Argument Pattren of Control Group																Total
	-	+	+-	-, 0	+, 0	+, -	+, +	+-, 0	+-, -	+-, +	+-, 0, 0	+-, +-, +	-, 0, 0, 0	Assumption	Constraint	Risk	None	
A																	14	14
B	3	10	6														9	28
C				2	3			1			1		1				9	17
D						5	2		3	3		1						14
E														5	23	2		30
Grand Total	3	10	6	2	3	5	2	1	3	3	1	1	1	5	23	2	32	103

Table 16. Simplification of pattern variant of Test group

Type	Simplify Argument Pattern of Test Group																					Total
	-	+	+-	-, 0	+, 0	+, -	+, +	+-, 0	+-, -	+, -, 0	+, -, -	+, +, -	+-, -, 0	-, 0, 0, 0	+-, -, 0, 0	+-, +-, -, -	+, -, -, 0, 0	Assumption	Constraint	Risk	None	
A																					17	17
B	1	9	5																		12	27
C				3	6			3		1			1	1	1		1				17	34
D						5	2		4		1	1				1						14
E																		14	16	3		33
Grand Total	**1**	**9**	**5**	**3**	**6**	**5**	**2**	**3**	**4**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**14**	**16**	**3**	**46**	**125**

5.5 Measuring the Pattern Variant

To explore the pattern variant and get more insight from the data, we conduct a quantitative data analysis using a paired samples t-test. This type of test is used to compare groups that are related in some way. In our case, the matched are the attributes of each pattern, for example the pattern of positive (+) in the control and the control group. The step that we conduct as follows:

1. The first step we state the null and the alternative hypothesis.

   Ho: $\mu 1=\mu 2$, (there are no difference between the means)

   Ha: $\mu 1\neq\mu 2$, (the population means is different)

   For this analysis, the significance level is 0.05, and the test method is paired sample t-test.

2. The data that we use are generated from Table 15 and Table 16 with small addition to the missing pattern. The addition is, if one of the patterns of the control group is not presented in the test group then the value of the missing pattern becomes zero in the test group, this addition apply two ways (vice versa). For example, the pattern +-, 0, 0 of the control group is not present in the test group, then that pattern value in the test group become zero (0). The result data presented as follows:

Control Group			Test Group		
Type	Attribute	Count	Type	Attribute	Count
A	None	14	A	None	17
B	-	3	B	-	1
	+	10		+	9
	+-	6		+-	5
	None	9		None	12
C	-, 0	2	C	-, 0	3
	+, 0	3		+, 0	6
	+-, 0	1		+-, 0	3
	+, -, 0	0		+, -, 0	1
	+-, -, 0	0		+-, -, 0	1
	+-, 0, 0	1		+-, 0, 0	0
	-, 0, 0, 0	1		-, 0, 0, 0	1
	+-, -, 0, 0	0		+-, -, 0, 0	1
	+, -, -, 0, 0	0		+, -, -, 0, 0	1
	None	9		None	17
D	+, -	5	D	+, -	5
	+, +	2		+, +	2
	+-, -	3		+-, -	4
	+-, +	3		+-, +	0
	+, -, -	0		+, -, -	1
	+, +, -	0		+, +, -	1
	+-, +-, +	1		+-, +-, +	0
	+-, +-, -, -	0		+-, +-, -, -	1
E	Assumption	5	E	Assumption	14
	Constraint	23		Constraint	16
	Risk	2		Risk	3
Total		103	Total		125

3. Then we calculate the data:

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	3.961538462	4.807692308
Variance	28.67846154	31.84153846
Observations	26	26
Pearson Correlation	0.84690462	
Hypothesized Mean	0	
df	25	
t Stat	-1.412114456	
P(T<=t) one-tail	0.08512175	
t Critical one-tail	1.708140745	
P(T<=t) two-tail	0.1702435	
t Critical two-tail	2.059538536	

4. Result of the calculation shown that P-value was 0.170, and it is above the significant level of 0.05. Therefore, we accept the Ho, which mean that there is no difference between the mean of populations.

After conducting paired sample t-test, we found that there was no significant difference in the number of pattern made by the control group (M = 3.96, SD = 5.36) and the test group (M = 4.80, SD = 5.64); $t(25) = -1.412$, p= .170. These results suggest that the treatment does not have an effect on number of pattern variants made by designers.

5.6 Conclusion

Using the event-listing matrix, we present the flow from the time that designers identify design issue until they made a design decision. Furthermore, using this matrix we can identify five types of rationale that designers made during the design discourse session.

We also explore the types in more detail and identify the pattern variance. Then we use paired samples t-test to measure the effect of treatment to this pattern variant of the control group and the test group. However, the result is not significant, which mean that the variant of pattern that designers made is not influenced by the treatment that was given.

6 Process Mining

In this thesis, we also use process mining to help us analyze and help identify the product of the different techniques. The objective of process mining is to acquire the extraction of related-process information with the use of events from business processes (Van der Aalst, 2011), or in other words extracting knowledge from event logs. The knowledge that comes out is in the form of process models, which can be used further for analyzing or optimizing processes.

Because we have explicitly code the different techniques and the Design Rationale elements with the time that they occur during the design discourse session, we can conduct process mining to analyze the reasoning process and their product. Furthermore, we can link these products with the process. This is done to ensure that the reader has insight in the statements on which our conclusions have been based.

6.1 Tool

Implementing process mining in our research, we use Disco by Fluxicon (https://fluxicon.com/disco) as our tool. This tool base on the Fuzzy Mining (Günther & Van Der Aalst, 2007). It is an approach of process mining to overcome the "spaghetti-like" problem, the problem that the model shows all details without distinguishing what is important and what is not.

We consider this tool is user friendly because it can create process models directly from our data. With this tool, we also can define our desired level of abstraction and create filters directly from the techniques of design reasoning or the Design Rationale elements that we obtain from the codes.

6.2 Creating the Process Models

Generating the process model from our data, we conduct the following step:

1. **Preparing the data**, the first step is collecting all the codes, the time that they occur, and the team that they belong to. Next step is combining each code, time, and team number as a single event log. And then we organize the event

logs into two major files, the control groups and the test group. The outputs of this step are two CSV files, Table 17 displays the example of this dataset files.

Table 17 Dataset of event logs

Team No	Time	Codes	Date	Combine
Team 2	18:54.6	Assumption Analysis	12/14/15	12/14/15 0:18:55
Team 2	31:04.3	Assumption Analysis	12/14/15	12/14/15 0:31:04
Team 2	06:35.3	Constraint Analysis	12/14/15	12/14/15 0:06:35
Team 2	26:05.4	Constraint Analysis	12/14/15	12/14/15 0:26:05
Team 2	16:06.8	Constraint Analysis	12/14/15	12/14/15 1:16:07
Team 2	37:27.5	Option Generation	12/14/15	12/14/15 0:37:27
Team 2	02:33.2	Problem Structuring	12/14/15	12/14/15 0:02:33
Team 2	06:00.0	Problem Structuring	12/14/15	12/14/15 0:06:00
Team 2	08:25.1	Problem Structuring	12/14/15	12/14/15 0:08:25
Team 2	13:18.1	Problem Structuring	12/14/15	12/14/15 0:13:18

2. **Input the data**, to create the process model, the tool need three major categories for their inputs, the Case ID, the Activity, and the Time Stamp. Therefore, we arrange our data base on this input categories, we use the team number as the Case ID, the design reasoning and Design Rationale elements as the Activity, and the time as the Time Stamp. Figure 6.1 shows the process of importing the data.

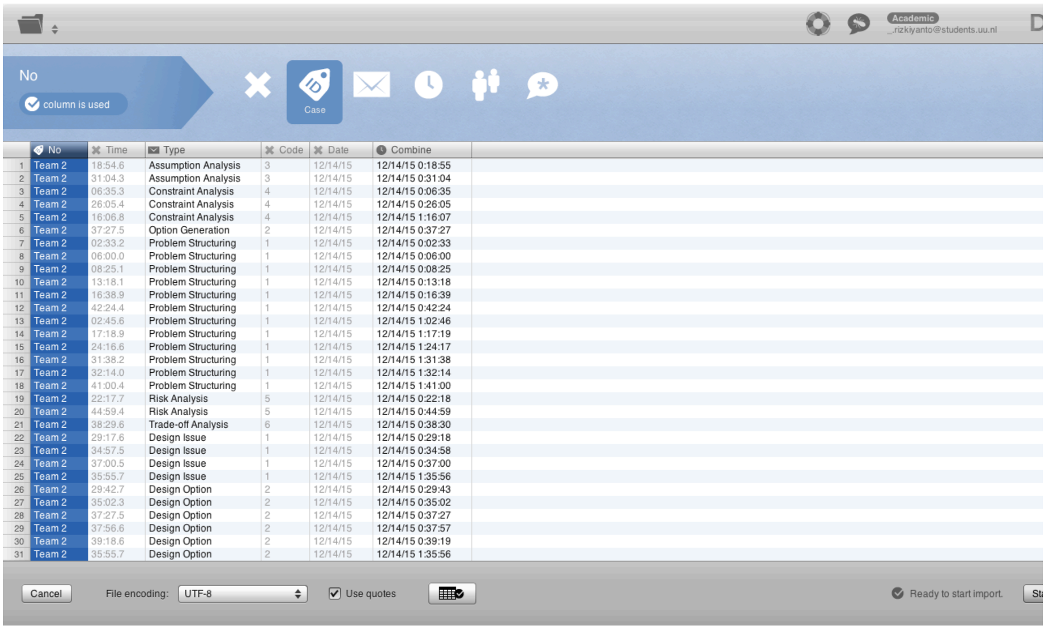


Figure 6.1 Importing data to Disco

3. **Create the process model**, the process model is automatically created after importing the data. To help analyze the data, the tool provides filters to set the data configuration for the desired process models. For example, we can set the

filter to find out the process of design reasoning technique of just one team or the whole group. Furthermore, we can also set the filter to show or hide the elements of Design Rationale, so we can analyze their link to the process on which they come out from. Figure 6.2 (below) shows the complete process mining of the control group as the output of the tool. However, because the model is really huge, to see the caption of design resoning techniques and the Design Rationale need to zoom the model directly, that is why we add annotation and arrow to point each elements that is present in the model. The process model of each team and group in detail available in Appendix D.

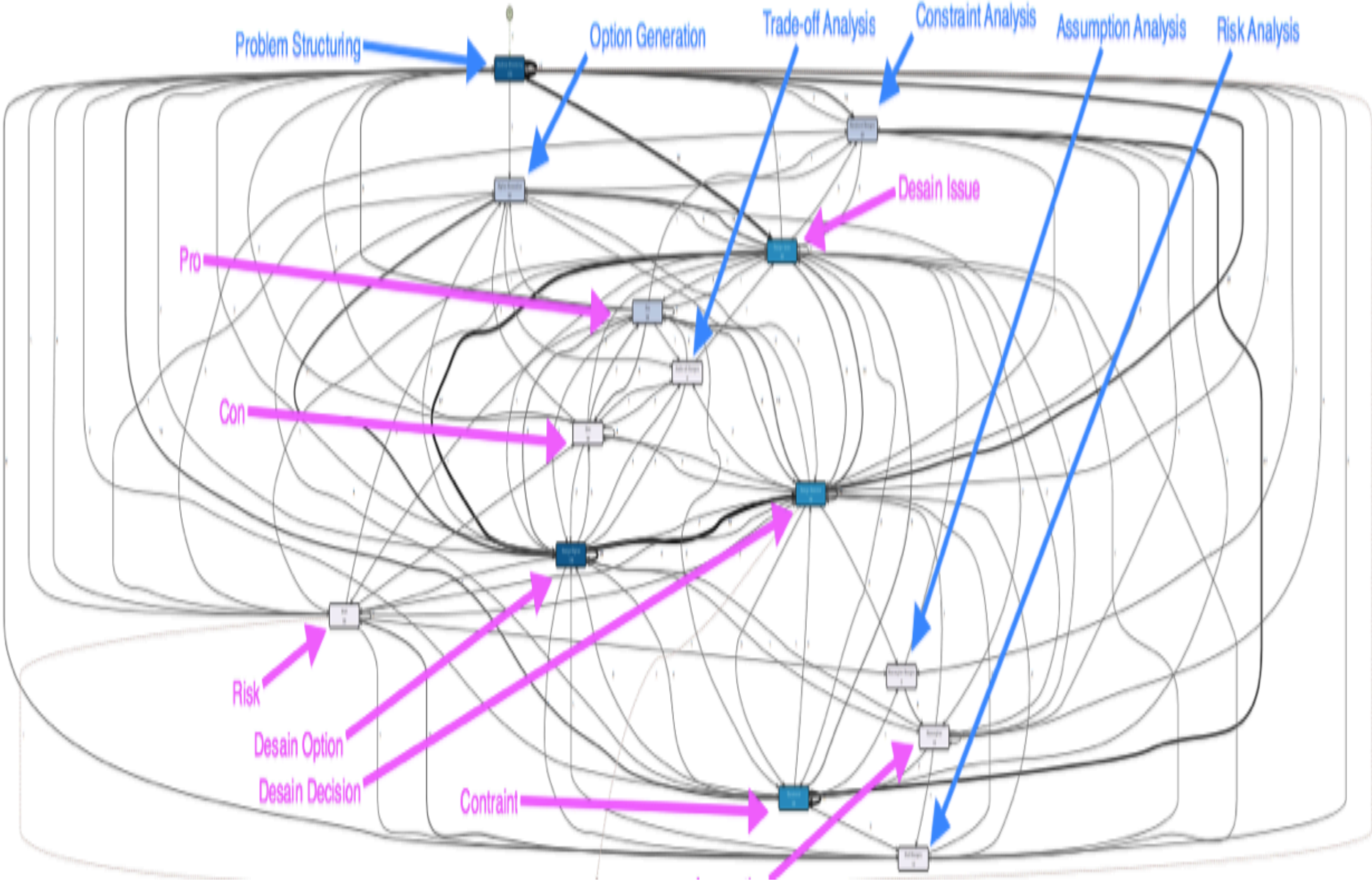


Figure 6.2 The complete process model of control group

6.3 Process Model Simplification

Before we conduct analysis on the process model, the tool provides a mechanism to simplify the complex process model base on fuzzy mining algorithm (Günther & Van Der Aalst, 2007). This was the first mining algorithm to introduce the "map metaphor" to process mining, which includes advanced features like seamless process simplification and highlighting of frequent activities and paths. These features allow us to adjust the detail that we want so see from the process model.

The tool provide two sliders that user can use to modify the level of detail that is shown in process model, they are:

1. Activities slider, this slider influences the number of activities shown in the process model, ranging from only the most frequent until all activities including the least frequent activities.
2. Paths slider, this slider determine on how many paths is shown in the process model, ranging from the most dominant path flow until all connection between the activities.

Figure 6.2 (above) shows the configuration of the slider with 100 % activities and 100 % paths, which resulting in "spaghetti-like" model. For further analysis, we used the paths slider to simplify our process model. As an example Figure 6.3 (below) show the simplification of the complete process model of the control group using 100 % activities and 10 % paths.

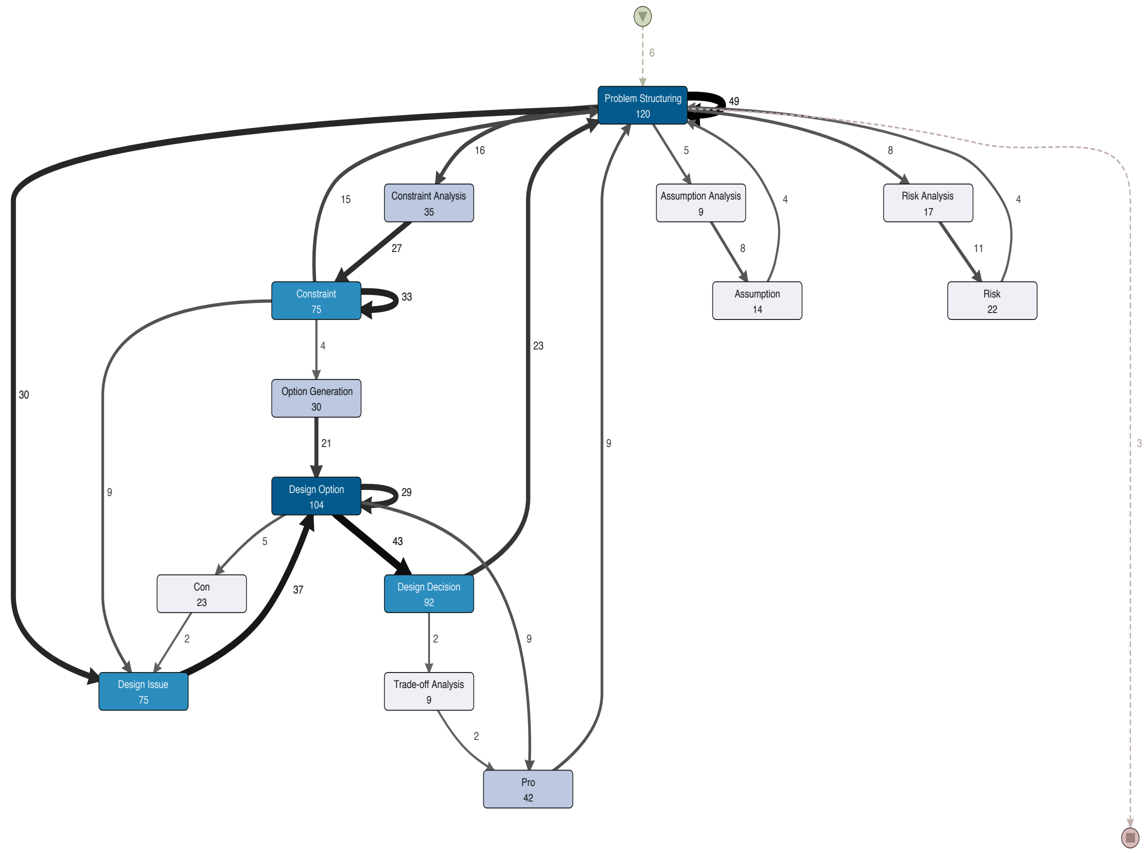


Figure 6.3 The simplification result of complete process model of control group

6.4 Process Model Analysis

We analyze the process model by the distinction of that we made, and that is the control group and the test group. For each group we made two models, first the complete model that presents the design reasoning techniques and the Design Rationale elements, within this model we can analyze the relation between the process (design reasoning techniques) and the product (Design Rationale elements).

The second model is the process model that consist only the design reasoning techniques, this model can help us understand the flow of the process that designers made during the design discourse session.

6.4.1 First Model of Control Group

For this model we use 100 % activities and 10 % paths, the result is presented with Figure 6.4 (below).

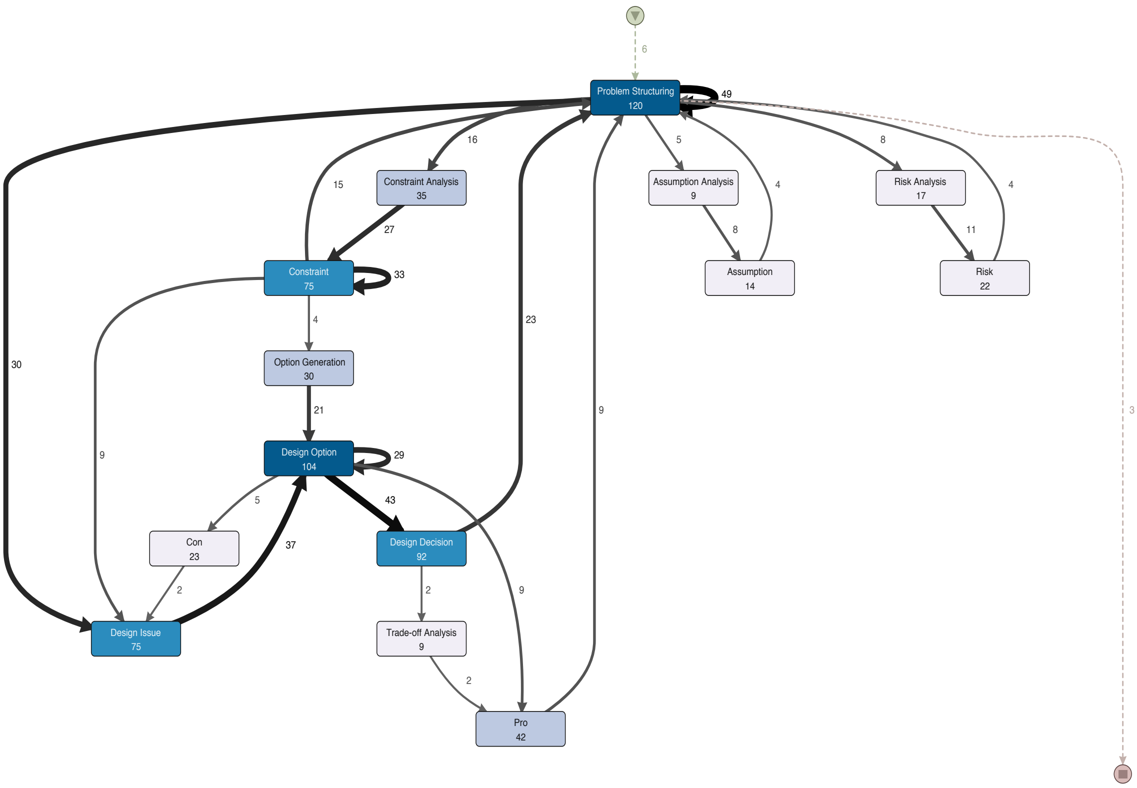


Figure 6.4 Process model of control group with 100% activities and 10% paths

From this process model we discovered that:

1. Most of Design Issues are created after Problem Structuring. This means that designers identify most of this issue after they structure the problem.
2. There are a clear flow of Design Issue to Design Option and then goes to Design Decision.
3. Assumption, Constraint, and Risk are preceded by their analysis. For example, Assumption Analysis always followed by Assumption.
4. One thing interesting is that the Pro and Con majority come after Design Option and not by Trade-off Analysis. We consider this because Trade-off Analysis is due to comparing two or more Design Option with showing their

Pro and Con, while in the design discourse session there are some arguments of Pro and Con for Design Option that is stand-alone.

6.4.2 First Model of Test Group

The first model with complete process model of test group presented with Figure 6.5 (below). This process model displays all the design reasoning techniques and the Design Rationale elements from all the team in the test group. For this model we also use 100 % activities and 10 % paths as the configurations.

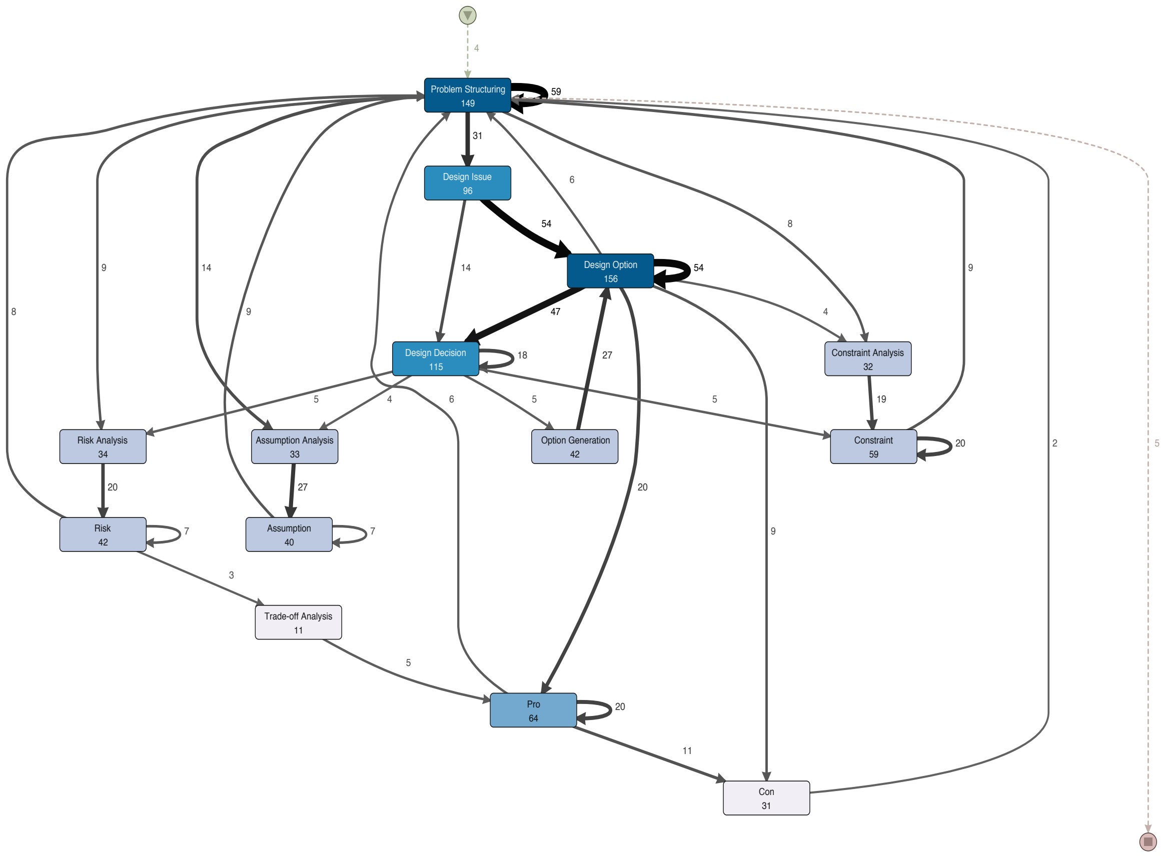


Figure 6.5 Process model of test group with 100% activities and 10% paths

From this process model we discovered that:

1. Most of Design Issue are created after Problem Structuring
2. There are also a clear flow of Design Issue to Design Option and then goes to Design Decision.

3. Assumption, Constraint, and Risk are preceded by their analysis. For example, Assumption Analysis always followed by Assumption.
4. The majority of the Trade-off Analysis followed by the positive argument (pro), then the designers continue with the negative argument (cons).

6.4.3 Second Model of Control Group

The second model we use 100% activities and 30% paths as the configuration. The process model is shown in Figure 6.6 (below).

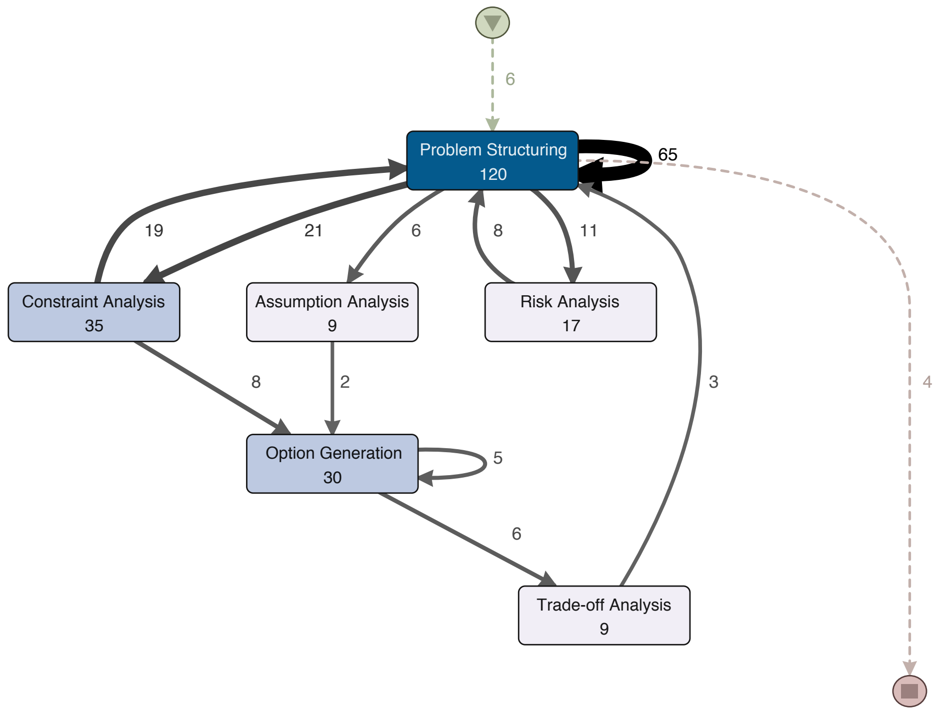


Figure 6.6 Process model of reasoning techniques of control group

This process resembles of "lasagna-like" process, which means that it have a clear structure. Furthermore, we discovered that:

1. There are four major stages in the process flow. The Problem Structuring is at the first stage. The second stage consists of Constraint Analysis, Assumption Analysis, and Risk Analysis. Option Generation available at the third stage and followed by Trade-off Analysis at the last stage.

2. Problem Structuring followed by Constraint Analysis, Assumption Analysis, or Risk Analysis. Before creating Option Generation, the designers usually made Constraint Analysis or Assumption Analysis. Trade-off Analysis is possible after designers execute Option Generation. We consider this is natural, because Trade-off Analysis is comparing between two or more available Design Option that is created by the Option Generation.

6.4.4 Second Model of Test Group

For the second model of test group that show only the design reasoning techniques we use 100% activities and 90% paths. The process model is shown bellow (Figure 6.7).

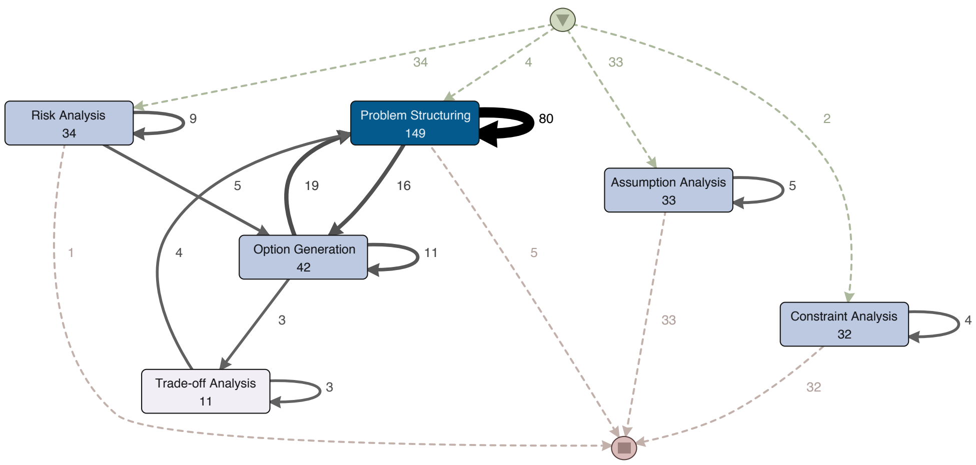


Figure 6.7 Process model of reasoning of test group with 100% activities and 90% paths

From the process model of reasoning technique of test group, we found that:

1. The stage of reasoning process is not very clear. We presume this is due to several teams start their discussion not with problem structuring, because the treatment prompts them with several techniques that are available to conduct their design discussion.
2. There is a visible flow from Risk Analysis to Option Generation, from Option Generation to Problem Structuring.

3. The Option Generation followed by the Option Generation, we can assume this happens when designers have two or more Design Option, and then they use the Trade-off Analysis to compare between these choices.

6.5 Identify the Correlation

Base on the analysis of the process model above, we can conclude that there were identified flows between:

1. Problem Structuring to Design Issue
2. Option Generation to Design Option
3. Assumption Analysis to Assumption
4. Constraint Analysis to Constraint
5. Risk Analysis to Risk

Using the data from Table 5 (Page 27) and Table 6 (Page 27), we analyze the correlation of this process (design reasoning techniques) to the product (Design Rationale elements) one by one.

6.5.1 Problem Structuring to Design Issue

A Pearson product-moment correlation coefficient was computed to assess the relationship between the Problem Structuring and the Design Issue. There was a positive correlation between the two variables, $r = 0.185$, $n = 12$. A scatterplot summarizes the results (Figure 6.8). Overall, there was a weak, positive correlation between Problem Structuring and Design Issue. Increases in Problem Structuring were correlated with increases in Design Issue.

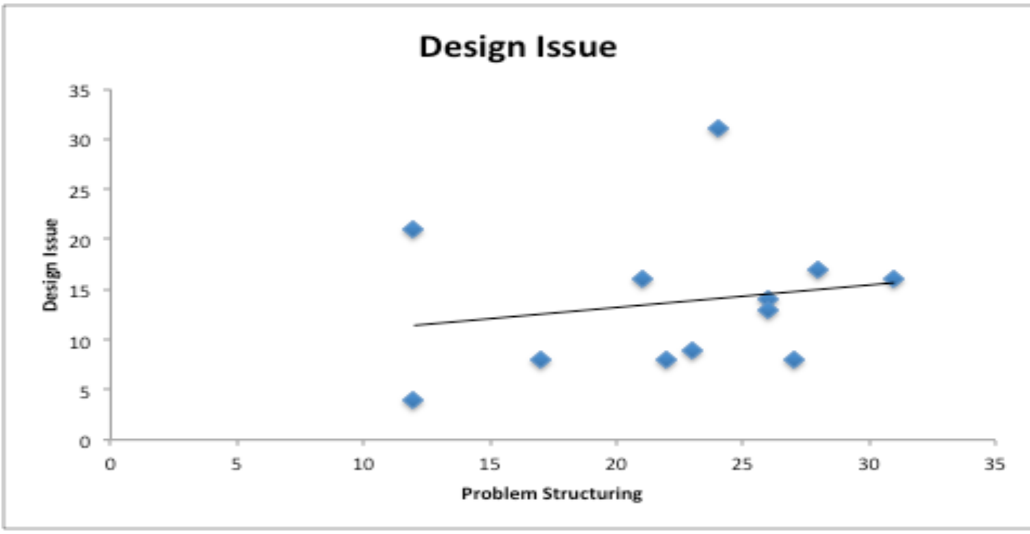


Figure 6.8 Problem Structuring to Design Issue

6.5.2 Option Generation to Design Option

A Pearson product-moment correlation coefficient was computed to assess the relationship between the Option Generation and the Design Option. There was a positive correlation between the two variables, r = 0.730, n = 12. A scatterplot summarizes the results (Figure 6.9). Overall, there was a strong, positive correlation between Option Generation and Design Option. Increases in Option Generation were correlated with increases in Design Option.

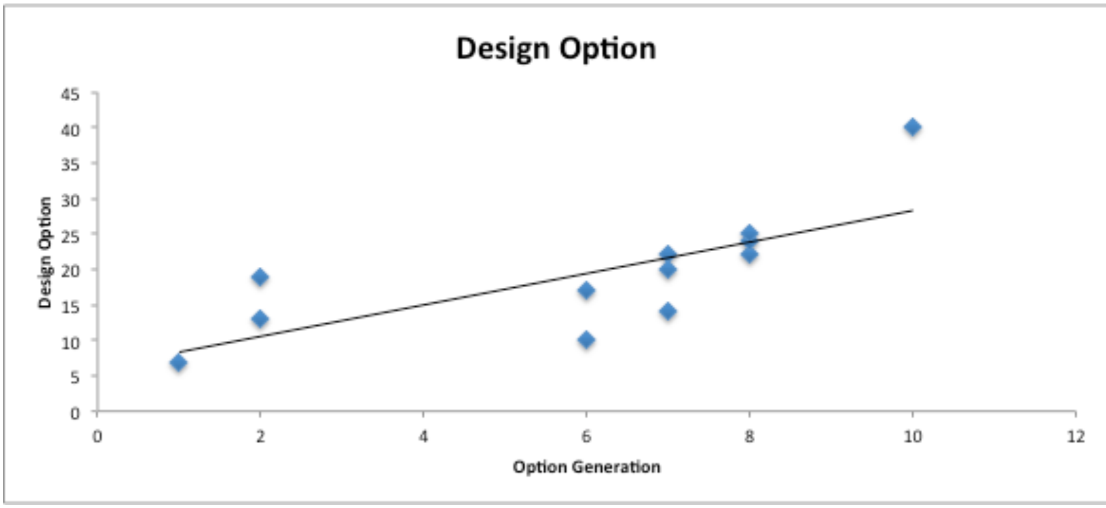


Figure 6.9 Option Generation to Design Option

6.5.3 Assumption Analysis to Assumption

A Pearson product-moment correlation coefficient was computed to assess the relationship between the Assumption Analysis and the Assumption. There was a positive correlation between the two variables, r = 0.866, n = 12. A scatterplot summarizes the results (Figure 6.10). Overall, there was a strong, positive correlation between Assumption Analysis and Assumption. Increases in Assumption Analysis were correlated with increases in Assumption.

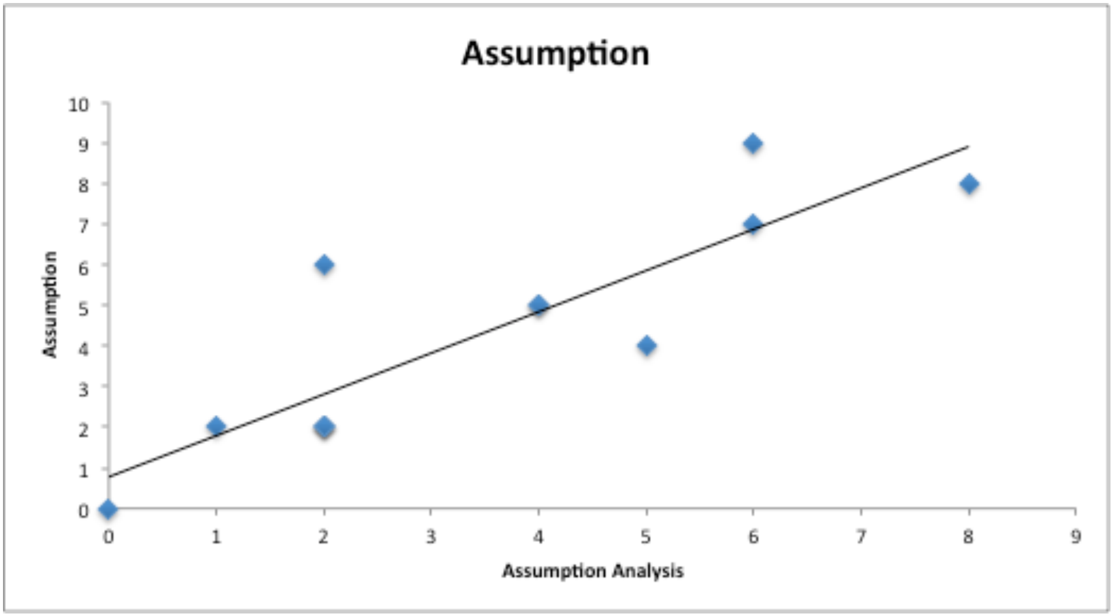


Figure 6.10 Assumption Analysis to Assumption

6.5.4 Constraint Analysis to Constraint

A Pearson product-moment correlation coefficient was computed to assess the relationship between the Constraint Analysis and the Constraint. There was a positive correlation between the two variables, r = 0.540, n = 12. A scatterplot summarizes the results (Figure 6.11). Overall, there was a strong, positive correlation between Constraint Analysis and Constraint. Increases in Constraint Analysis were correlated with increases in Constraint.

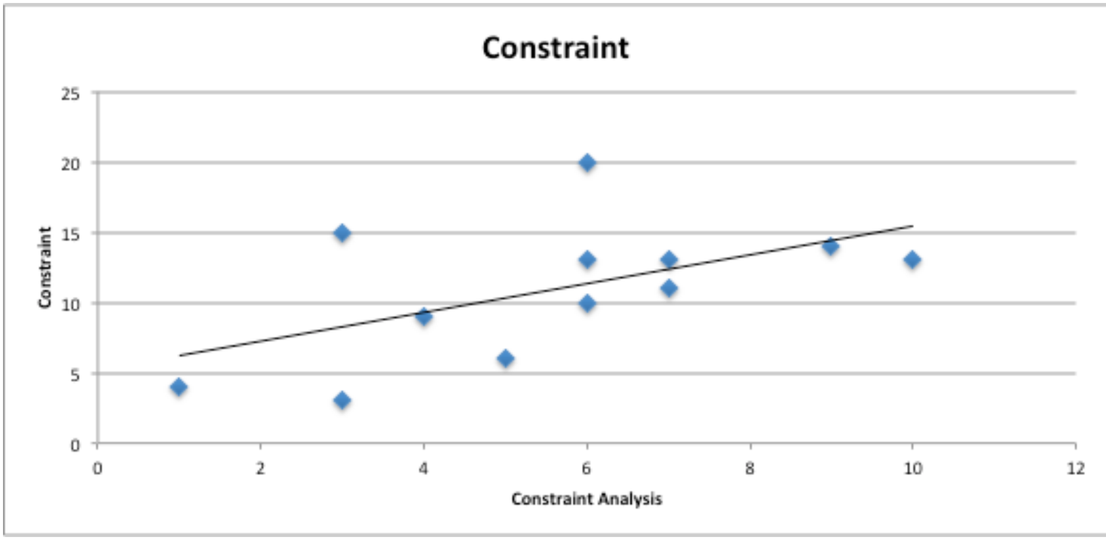


Figure 6.11 Constraint Analysis to Constraint

6.5.5 Risk Analysis to Risk

A Pearson product-moment correlation coefficient was computed to assess the relationship between the Risk Analysis and the Risk. There was a positive correlation between the two variables, r = 0.649, n = 12. A scatterplot summarizes the results (Figure 6.12). Overall, there was a strong, positive correlation between Risk Analysis and Risk. Increases in Risk Analysis were correlated with increases in Risk.

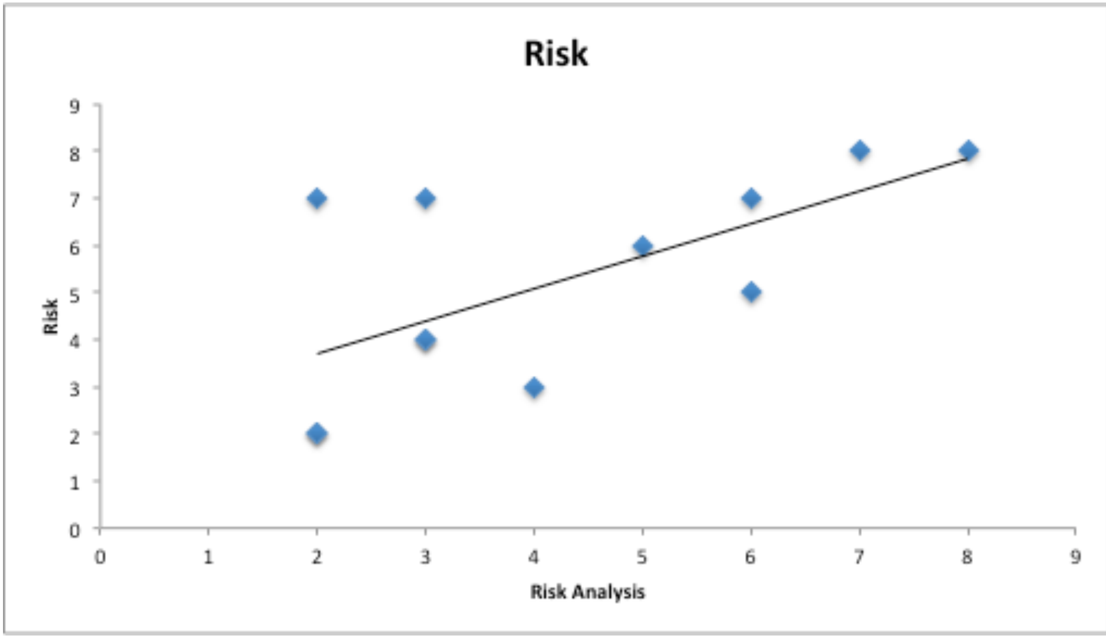


Figure 6.12 Risk Analysis to Risk

6.6 Conclusion

The process mining helps us create the process model that comes out from each group. Base on this process model we can identify and analyze the flow of the process (design reasoning technique) to the product (Design Rationale elements). The model also indicates the majority flow that appears. Furthermore, because we split the process between control group and test group, we could discover the structure of reasoning process of each group.

Using the structure of the process to the product, then we can calculate their correlation using Pearson product-moment correlation coefficient which results in strong, positive correlation for Option Generation to Design Option, Assumption Analysis to Assumption, Constraint Analysis to Constraint, and Risk Analysis to Risk. However, we found weak, positive correlation between Problem Structuring and Design Issue. This happens because the designer use more one or more of the Problem Structuring techniques before identifies the Design Issue.

7 Result

In the previous section, we have analyzed the results of the experiment. We create the event-listing matrix from the first experiment data and use this matrix to identify the type of Design Rationale element pattern and their variance. Furthermore, we also use process mining to understand the process and the product as an output of the design reasoning process. In summary, we have several findings and they are discussed in the following sections.

7.1 Finding 1 – The Reflective Method Intensify the Frequency of Design Reasoning Technique and the Design Rationale Element

We observed that the amount of design reasoning technique and the Design Rationale elements from the first experiment. The treatment that we use was the design reasoning reflective method (Schriek, 2016), they are meant to prompt the designers to question their decisions in order to come to better thoughtful design, and to help them with several possible reasoning techniques for making that decision.

We discovered that the amount of design reasoning technique of the test group (with treatment) more than the control group (without treatment). The evident form Table 18 shows the frequency of the design reasoning technique during the design discourse session. The result reveals that the test group produces 37 % more than the other group.

Table 18 Design reasoning frequency

Reasoning Technique	Control Group						Total	Average	Test Group						Total	Average
	C1	C2	C3	C4	C5	C6			T1	T2	T3	T4	T5	T6		
Assumption Analysis	2	0	2	2	1	2	9	1.50	8	4	6	4	6	5	33	5.50
Constraint Analysis	3	4	9	10	6	3	35	5.83	5	6	1	6	7	7	32	5.33
Option Generation	1	6	2	8	7	6	30	5	10	2	7	8	7	8	42	7
Problem Structuring	12	27	31	12	21	17	120	20	24	23	26	28	22	26	149	24.83
Risk Analysis	2	4	3	3	2	3	17	2.83	5	7	2	8	6	6	34	5.67
Trade-off Anaysis	1	3	0	5	0	0	9	1.5	3	0	1	4	3	0	11	1.83
Total	21	44	47	40	37	31	220	36.67	55	42	43	58	51	52	301	50.17
Average	3.50	7.33	7.83	6.67	6.17	5.17	36.67	6.11	9.17	7.00	7.17	9.67	8.50	8.67	50.17	8.36

While from Table 19 displays that the frequency of the Design Rationale elements of the test group produces 30 % more than the control group. It shows that team that using large amount of design reasoning technique during the design discourse session also produce the amount of Design Rationale elements.

Table 19 Design Rationale elements frequency

DR Element	Control Group						Total	Average	Test Group						Total	Average
	C1	C2	C3	C4	C5	C6			T1	T2	T3	T4	T5	T6		
Assumption	2	0	2	6	2	2	14	2.33	8	5	7	5	9	4	38	6.33
Con	2	3	0	11	2	4	22	3.67	11	1	5	10	3	1	31	5.17
Constraint	3	9	14	13	20	15	74	12.33	6	13	4	10	13	11	57	9.50
Design Decision	4	8	16	21	16	8	73	12.17	31	9	14	17	8	13	92	15.33
Design Issue	4	8	16	21	16	8	73	12.17	31	9	14	17	8	13	92	15.33
Design Option	7	10	19	22	22	17	97	16.17	40	13	20	24	14	25	136	22.67
Pro	7	7	9	16	2	3	44	7.33	22	7	11	13	5	3	61	10.17
Risk	2	3	4	4	2	7	22	3.67	6	8	7	8	5	7	41	6.83
Total	31	48	80	114	82	64	419	69.83	155	65	82	104	65	77	548	91.33
Average	5.17	8.00	13.33	19.00	13.67	10.67	69.83	11.64	25.83	10.83	13.67	17.33	10.83	12.83	91.33	15.22

To get more insight into the data, we conducted a 2-sample t-test for each design reasoning techniques and Design Rationale elements in Section 4.5. We found that the treatment with the reflective method of design reasoning had an effect on some techniques. The affected techniques are Assumption Analysis and Risk Analysis. Meanwhile, the elements of Design Rationale that affected by the treatment are Assumption and Risk.

7.2 Finding 2 – The Type of Design Rationale Element

Understanding the implicit Design Rationale that designers build with the design reasoning technique, we examined the pattern of the events. From the event-listing matrix, we classified that are five patterns type that designers build during the design discourse session, they are:

1. **Direct Decision**, this as the result of decision made straight from the issue.
2. **Focus Decision**, the designers discover the design option for the issue that they identified. However, they only focus on that option and overlook other possible option.
3. **Half consideration**, the designer develop more option to help them solve the issue. However, some options do not have an argument behind it.
4. **Full consideration**, designers develop several options that it is considered best at that moment support with some arguments behind it, and decide the best decision based on those arguments.
5. **No-link**, is not directly connected to any Design Issue or Design Option. This is possible because the argument is invalid or out of the context of the issue.

We also identify the variant of each pattern type, and conduct paired samples t-test in Section 5.5. However, measuring this pattern we found that they are no significance between the pattern variant of the control group and the pattern variant of the test group. In other words, the treatment has no significant impact in pattern variant that designers made during the design discourse session.

7.3 Finding 3 - Result of Process Mining

Understanding the link between the process and the product, we use process mining as our method. The process-mining tool that we use optimizes the Fuzzy mining algorithm (Günther & Van Der Aalst, 2007) to create and simplify the process model. From the simplify process models, we identified several flows that designers build during the design discourse session, they are:

1. Most of Design Issues are created after Problem Structuring. This means that designers identify most of this issue after they structure the problem.
2. There are clear flows of Design Issue to Design Option and then goes to Design Decision. This also emphasizes the flow that we made in the event-listing matrix to create Design Rationale elements pattern type (Finding 2).
3. The Assumption, Constraint, and Risk are preceded by their analysis. For example, Assumption Analysis always followed by Assumption.

We also measure the correlation using the link structure of the process to the product, we can calculate their correlation using Pearson product-moment correlation coefficient, and the flows are:

1. Problem Structuring to Design Issue
2. Option Generation to Design Option
3. Assumption Analysis to Assumption
4. Constraint Analysis to Constraint
5. Risk Analysis to Risk

The measurements result in strong, positive correlation for Option Generation to Design Option, Assumption Analysis to Assumption, Constraint Analysis to Constraint, and Risk Analysis to Risk. However, we found weak, positive correlation between Problem Structuring and Design Issue. This happens because the designers use more one or more of the Problem Structuring techniques before identifies the Design Issue.

8 Conclusion and Discussion

8.1 Conclusion

This research focuses on the design reasoning and Design Rationale elements in the design discourse session. The main research question is formulated as "*What is the link between design reasoning and Design Rationale in the design process?*"

In order to answer this question, four sub-questions have been drafted. The following section we provide the answer to all sub-questions and finally the answer to our main research question.

8.1.1 The Concept of Design Rationale and Design Reasoning (SQ1)

The first sub-question was formulated as follows: *"what are the central concepts of design rationale and what is design reasoning?"* From the literature study, we discover that the central concept of Design Rationale is focusing on the design decision and the elements that provide the justifications, the other alternatives, and the arguments behind the design decisions. All these elements captured in the framework of Design Rationale element, this framework is the refinement of several approaches to capture Design rationale.

Design reasoning itself is the process that designer use to come to the solution, that is the design decision. Researchers have identified several techniques to help designers made this decision; they are Problem Structuring, Option Generation, and Analysis of Assumption, Constraint, Risk, and Trade-off.

8.1.2 Quality Measure for Design Rationale (SQ2)

The second sub-question was formulated as follows: *"what are the existing quality measures for design rationale?"*. From our literature study, we found the measurements for requirements that designer could use to achieve a valid decision. However, we did not find any measurement for the concept of Design Rationale elements as a whole. Therefore, we propose the pattern type of Design Rationale elements. These patterns is the output of the event-listing matrix, the pattern type classifies on how the designers come to the decision base on the justifications, the other alternatives, and the arguments behind the design decisions. These patterns are:

(i) Direct Decision, (ii) Focus Decision, (iii) Half Consideration, (iv) Full Consideration, and (v) No-link. These patterns are the result of reasoning that designers made at the design discourse session. Although in the end, the quality of design decisions not just depend on the reasoning ability of the designers, but also affected by the knowledge and the creativity of the designers itself.

8.1.3 Rationale as the Product of Reasoning Process (SQ3)

The third sub-question was formulated as follows: "*what is the rationale that designers build during the reasoning process?"*. Using the process model, we identified the rationale that designer build. The process model helps us apprehended the common flows that appear from the reasoning process to the product. Using these flows, we calculate the correlation using the quantitative method, which results in strong, positive correlation for most of the techniques. Thus, increases in the number of reasoning techniques were correlated with increases in Design Rationale elements.

While focusing on the rationale, we can conclude that Problem structuring leads to Design Issue, Option Generation helps designers develop one or more Design Option, while the results of Assumption Analysis, Constraint Analysis, Risk Analysis, and Trade-off Analysis are positive or negative argument for the Design Option. Design Decision is the Design Option that designers think the best solution for the Design Issue. The pattern of this relation could be presented in five pattern types as we propose for the answer of sub-question two.

8.1.4 Improving Implicit Rationale (SQ4)

The fourth sub-question was formulated as follows: *"How does different design reasoning improve the implicit rationale?"* Using our first finding (section 7.1), we conclude that the treatment increases the amount of Assumption that designers made and the Risk that designers identified during the design discourse session. Therefore, by prompting the designers with the available design reasoning techniques could increase the quantity of argument that designers made. We see this as a way to improve the implicit rationale from the quantity aspect. However, for the quality of the argument itself still rely on the knowledge domain of the designers.

8.1.5 Answer to Main Research Question

The main research question is formulated as follows:

> "*What is the link between design reasoning and Design Rationale in the design process?*"

To identify the link between design reasoning and Design Rationale, this research has used the quantitative method, the qualitative method, and the process mining. The patterns of rationale that designers made during the design discourse session have been identified. The flow of process the product have been identified and measured. Base on these findings from previous sections and the answer of each sub-question in this section, we can conclude that the reasoning technique as the link to connect the design reasoning and Design Rationale in the design process.

8.2 Discussion

This research focuses on two approaches during design session; they are Design Rationale and the design reasoning. We have analyzed the link between these two methods. Our results reveal that the pattern of rationale that designers made with different reasoning technique. This has the implication on how to train and educate novice designers.

8.2.1 Limitation

Yin (2003) describes four different criteria for empirical research. These criteria are: construct validity, internal and external validity and reliability. We notice the limitation of this experiment related to the internal validity includes the student's background, student's experience, and their design abilities. To mitigate this, the students chose the team member themselves. While splitting the control group and the test group based on their grades for a previous assignment, so there is a mix of groups who have shown to do well, and those who have lesser grades.

Another limitation is that the identification of the reasoning process and the elements of Design rationale are subjective. To mitigate this, each transcript independently codified by two researchers. As noted in 4.3 the Cohen's Kappa Coefficient (Cohen, 1968) was 0.64, which indicated substantial agreement between the two coders. This method helped us ensure that the study is done with desirable quality.

Another limitation is that the design reasoning and elements of Design Rationale we observed were based on dialog from audio recordings. Non-verbalized exchanges such as pointing and looks could not be observed. We assume that in a group discussion, most of the considerations were communicated verbally.

Finally, another limitation was the number of participants is low, because the experiment depends on the number of students following the Software Architecture course.

8.2.2 Future Research

This thesis research has used the combination of qualitative and quantitative research method to explore the link between process and product of design discourse session. As mentioned earlier, the number of the participant for the experiment is low, with an addition of more participants the amount of data that could be analyzed would give more perspective. With these additional participants, a more quantitative test could be performed which result would be more comprehensive.

This research has shown the potential pattern of rationale to evaluate the type of design decision made by designers. Optimizing this pattern and combining with the measurement for the quality of design decision could be an interesting topic as the continuation of this project.

Finally, the treatment that used in the experiment was the reflective method of design reasoning with cards (Schriek, 2016). We think there is enough space to develop this approach in the future, which could help novice designers expanding their knowledge and improve their design skill.

References

Carroll, J. M., & Rosson, M. B. (1991). Deliberated evolution: Stalking the View Matcher in design space. *Human–Computer Interaction*, *6*(3-4), 281-318.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin 70 (4)*, 213.

Conklin, J., & Begeman, M. L. (1988). gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Information Systems (TOIS)*, *6*(4), 303-331.

Conklin, E. J., & Yakemovic, K. C. (1991). A process-oriented approach to design rationale. *Human-Computer Interaction*, *6*(3), 357-391.

De Neys, W. (2007). Implicit conflict detection during decision making. *Proceedings of the Annual Conference of the Cognitive Science Society*, Nashville, Tennessee, USA, 209-214.

Dutoit, A. H., McCall, R., Mistrík, I., & Paech, B. (2007). *Rationale management in software engineering*. Berlin: Springer Science & Business Media.

Dyer, M. (1988). Designing software for provable correctness: The direction for quality software. *Information and Software Technology*, *30*(6), 331-340.

Fischer, G., Lemke, A. C., McCall, R., & Morch, A. I. (1991). Making argumentation serve design. *Human–Computer Interaction*, *6*(3-4), 393-419.

Franklin, L. R.. (2005). Exploratory Experiments. *Philosophy of Science*, *72*(5), 888–899.

Günther, C. W., & Van Der Aalst, W. M. (2007). Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In *Business Process Management* (pp. 328-343). Berlin: Springer Heidelberg.

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998). The architecture tradeoff analysis method. *Proceedings Engineering of Complex Computer Systems, 1998. ICECCS'98. Fourth IEEE International Conference*, Monterey, CA, USA, 68-78.

Kunz, W., & Rittel, H. W. (1970). *Issues as elements of information systems* (Vol. 131). Berkeley, California: Institute of Urban and Regional Development, University of California.

Lee, J., & Lai, K. Y. (1991). What's in design rationale?. *Human–Computer Interaction*, *6*(3-4), 251-280.

Lee, J. (1997). Design rationale systems: understanding the issues. *IEEE intelligent systems*, *12*(3), 78-85.

MacLean, A., Young, R. M., Bellotti, V. M., & Moran, T. P. (1991). Questions, options, and criteria: Elements of design space analysis. *Human–computer interaction*, *6*(3-4), 201-250.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. California: Sage publications.

Paul, R., Niewoehner, R., & Elder, L. (2006). *The thinker's guide to engineering reasoning*. Foundation Critical Thinking, Dillon Beach, CA.

Parnas, D. L., & Clements, P. C. (1986). A rational design process: How and why to fake it. *Software Engineering, IEEE Transactions on*, *1*(2), 251-257.

Rationale. (2011). *Cambridge Advanced Learner's Dictionary & Thesaurus*. Retrieved December 5, 2015, from http://dictionary.cambridge.org/dictionary/english/rationale

Razavian, M., Tang, A., Capilla, R., & Lago, P. (2015). In Two Minds: How Reflections Influence Software Design Thinking.

Rittel, H. W., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, *4*(2), 155-169.

Rozanski, N., & Woods, E. (2012). *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Boston: Addison-Wesley.

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8-13.

Schriek, C. (2016). *Influencing design reasoning: a card game*. Unpublished manuscript.

Shum, S. B. (1995). Design argumentation as design rationale. *The encyclopedia of computer science and technology*, *35*(20), 95-128.

Shum, S. B., & Hammond, N. (1994). Argumentation-based design rationale: what use at what cost?. *International Journal of Human-Computer Studies*, *40*(4), 603-652.

Skoglund, M., & Runeson, P. (2009). Reference-based search strategies in systematic reviews. *Proceedings of the 13th international conference on Evaluation and Assessment in Software Engineering,* Swinton, UK, 31-40.

Tang, A. (2011). Software designers, are you biased?. *Proceedings of the 6th International Workshop on SHAring and Reusing Architectural Knowledge*, Waikiki, Honolulu, USA, 1-8.

Tang, A., Jin, Y., & Han, J. (2007). A rationale-based architecture model for design traceability and reasoning. *Journal of Systems and Software*, *80*(6), 918-934.

Tang, A., & Van Vliet, H. (2012). Design strategy and software design effectiveness. *Software, IEEE*, *29*(1), 51-55.

Tang, A., Tran, M. H., Han, J., & Van Vliet, H. (2008). Design reasoning improves software design quality. In S. Becker, F. Plasil, and R. Reussner(Eds.), *Quality of Software Architectures. Models and Architectures* (pp. 28-42). Berlin: Springer.

Toulmin, S. E. (1958). *The uses of argument*. Cambridge: Cambridge University Press.

UCI. (2010) Studying Professional Software Design. Retrieved from: http://www.ics.uci.edu/design-workshop/

Van Der Aalst, W. (2011). *Process mining: discovery, conformance and enhancement of business processes*. Berlin: Springer Science & Business Media.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Berlin: Springer Science & Business Media.

Yin, R. K. (2003). *Case study research: Design and methods*. California: Sage publications.

APPENDIX A

Documentation Template

SOFTWARE ARCHITECTURE

OF A

TRAFFIC SIMULATION SYSTEM

Group	XX
Names	Student 1 (XXX)
	Student 2 (XXX)
	Student 3 (XXX)

Contents

Product introduction

Describe in a few words (about a paragraph) on what the product to be designed is about.

Context Viewpoint

Use the following template for each of the views in the context viewpoint:

View: <name>

Model

Place here the model representation of the view

Description

Short description of the view

Glossary of elements

Give a description for each of the elements in the view

Id	Name	Description

Rationale

Describe shortly why the view is as it is. Notice that we want an argumentation about the view, and not an argumentation of why the view is included in the architecture description.

Functional Viewpoint

Use the following template for each of the views in the functional viewpoint:

View: <name>

Model

Place here the model representation of the view

Description

Short description of the view

Glossary of elements

Give a description for each of the elements in the view

Id	Name	Description

Rationale

Describe shortly why the view is as it is. Notice that we want an argumentation about the view, and not an argumentation of why the view is included in the architecture description.

Information Viewpoint

Use the following template for each of the views in the functional viewpoint:

View: <name>

Model

Place here the model representation of the view

Description

Short description of the view

Glossary of elements

Give a description for each of the elements in the view

Id	Name	Description

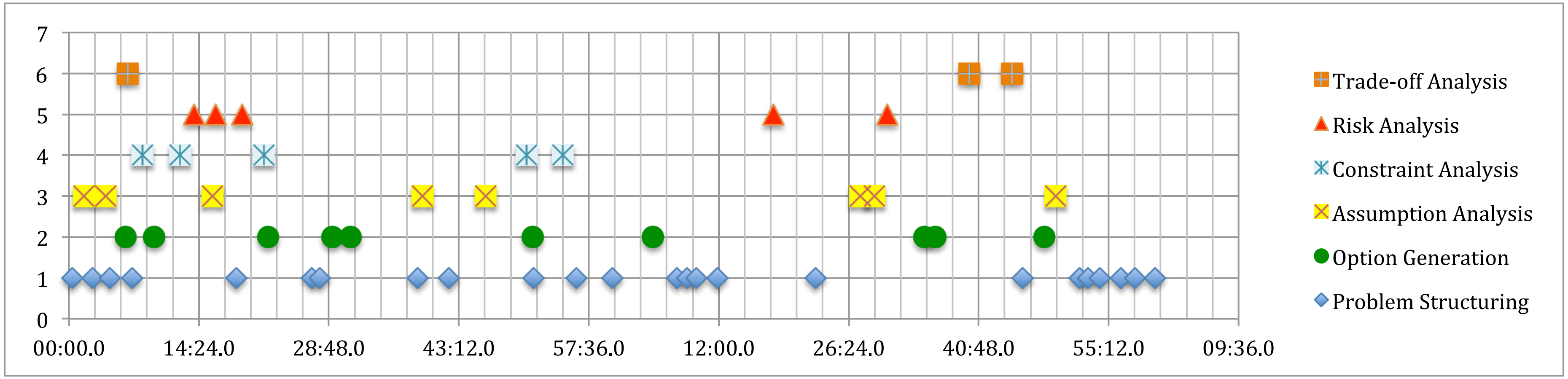
Rationale

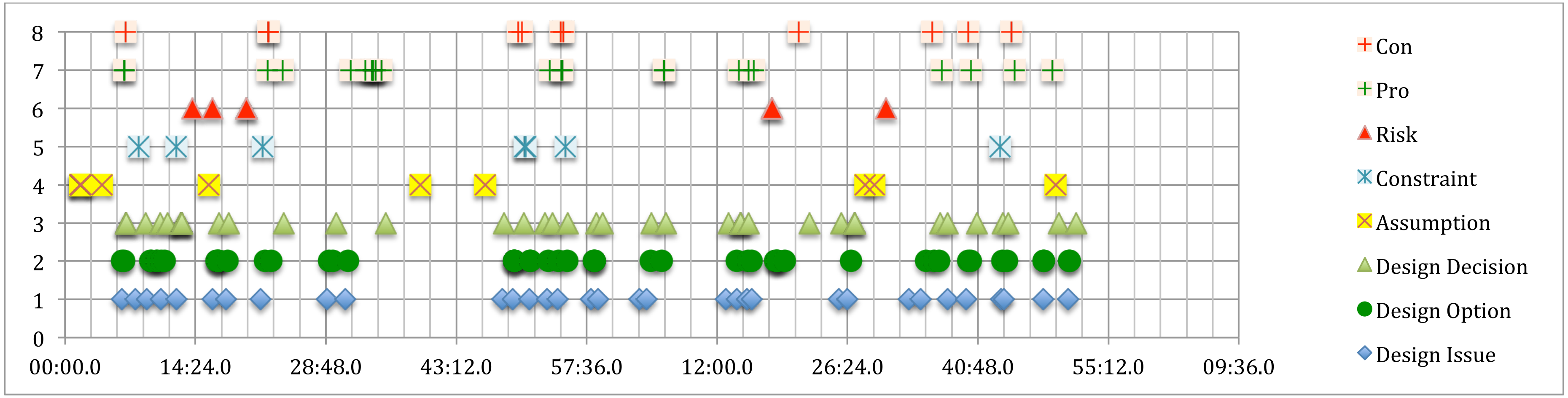
Describe shortly why the view is as it is. Notice that we want an argumentation about the view, and not an argumentation of why the view is included in the architecture description.

APPENDIX B

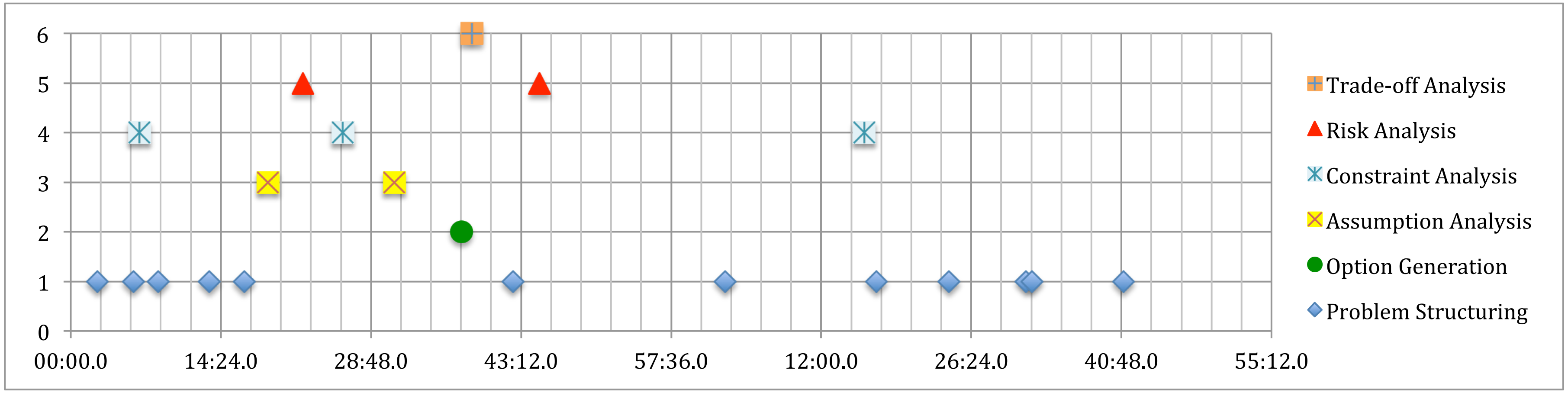
Time Plot Graph

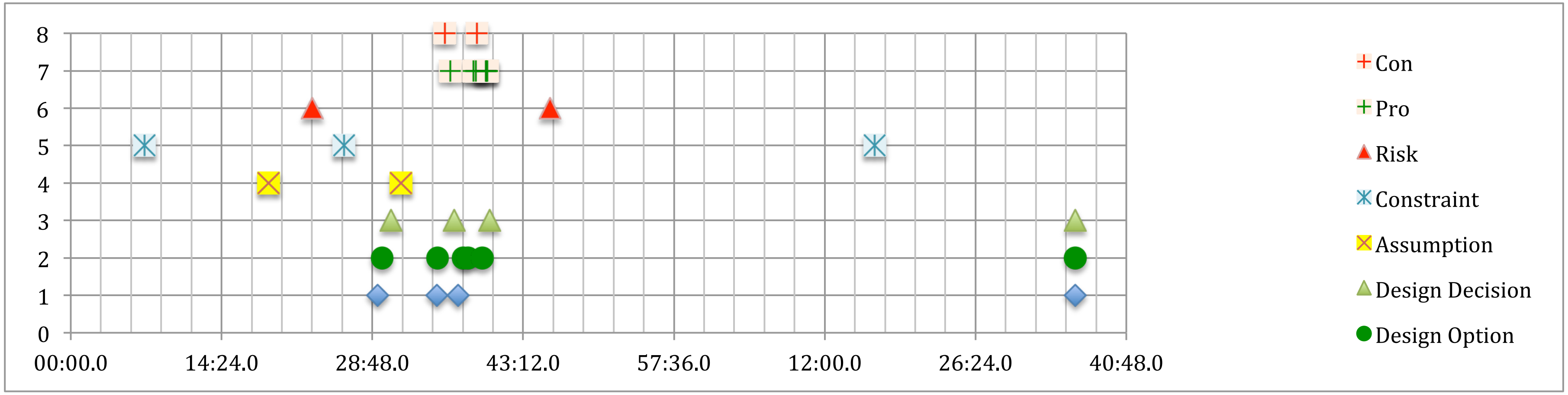
Time plot of design reasoning techniques and Design Rationale elements for Group 1



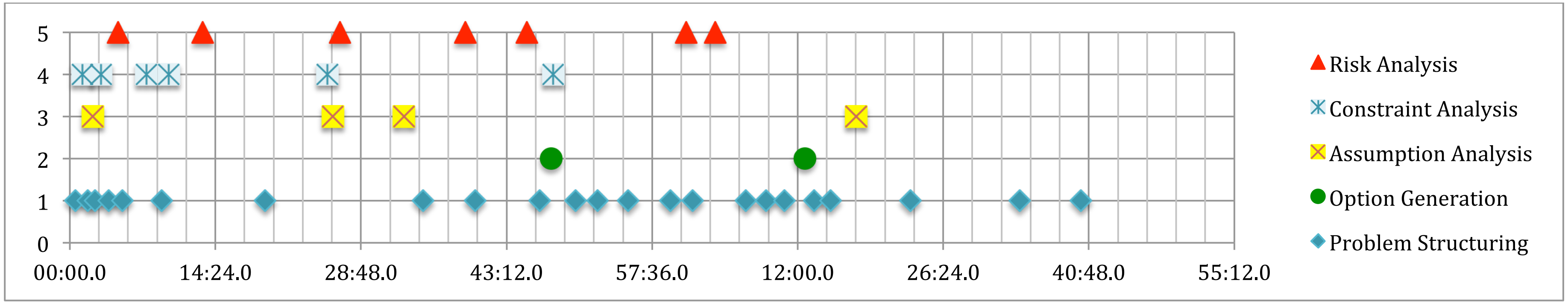


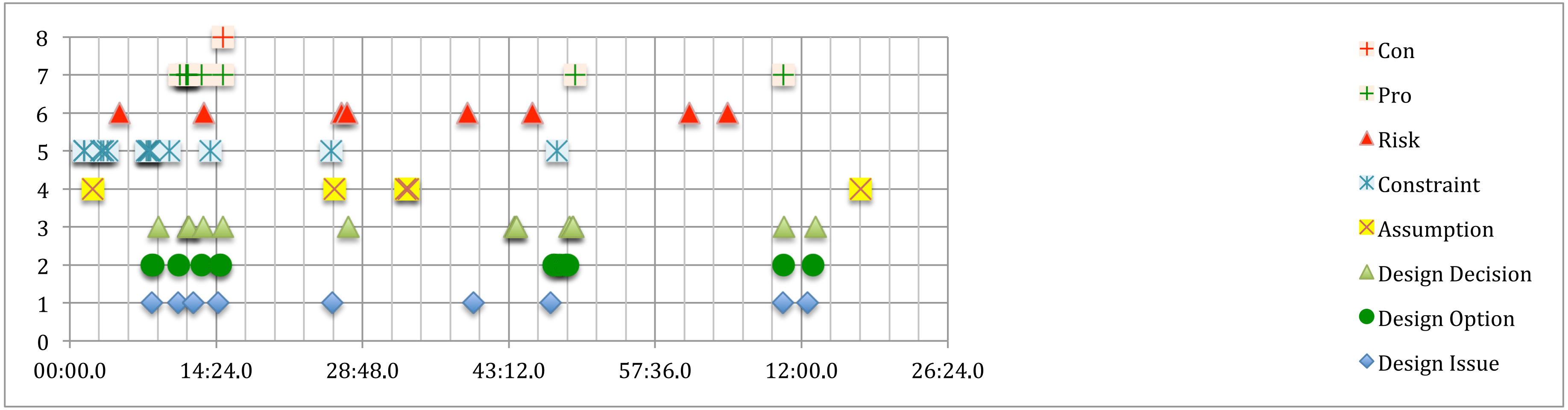
Time plot of design reasoning techniques and Design Rationale elements for Group 2



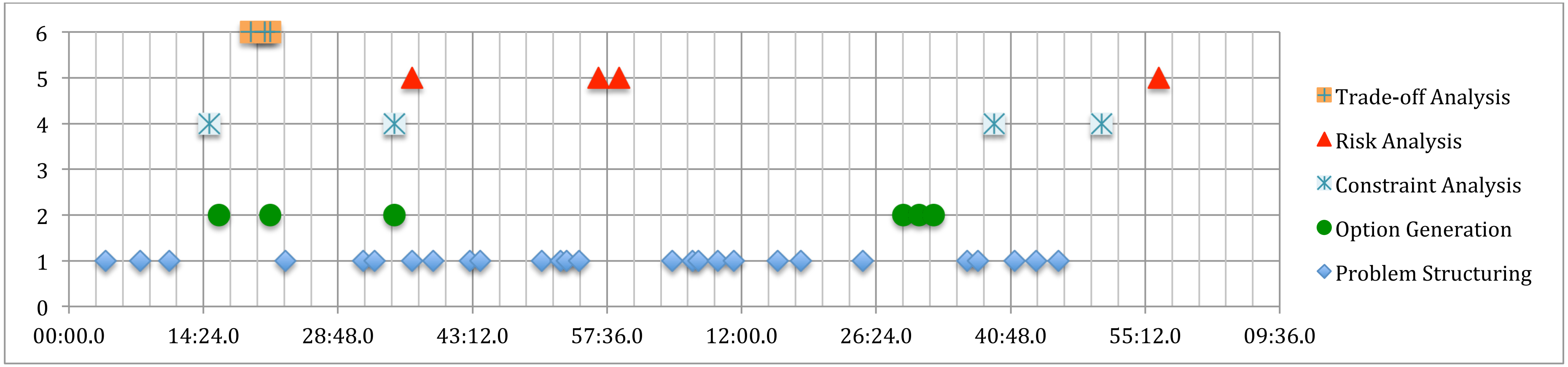


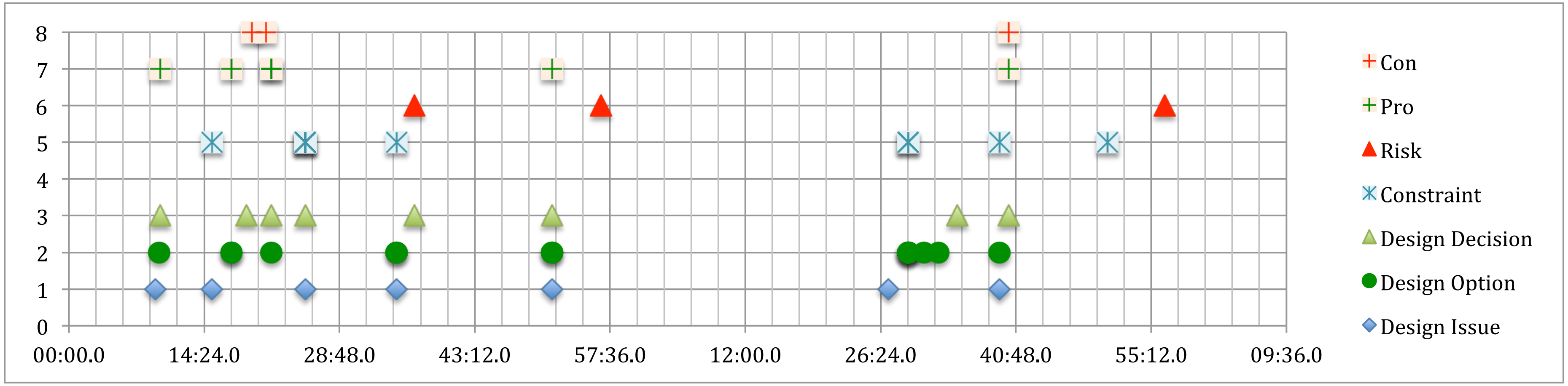
Time plot of design reasoning techniques and Design Rationale elements for Group 3



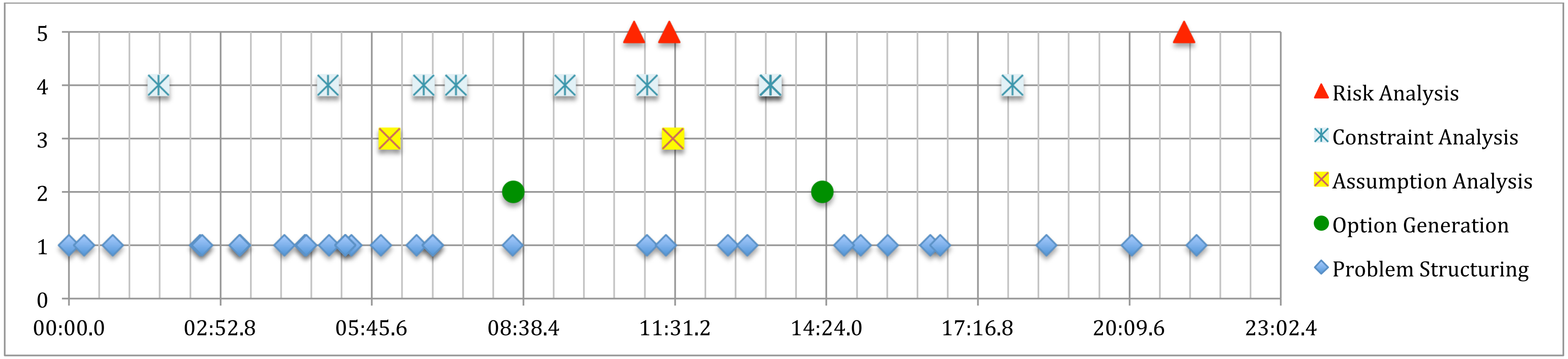


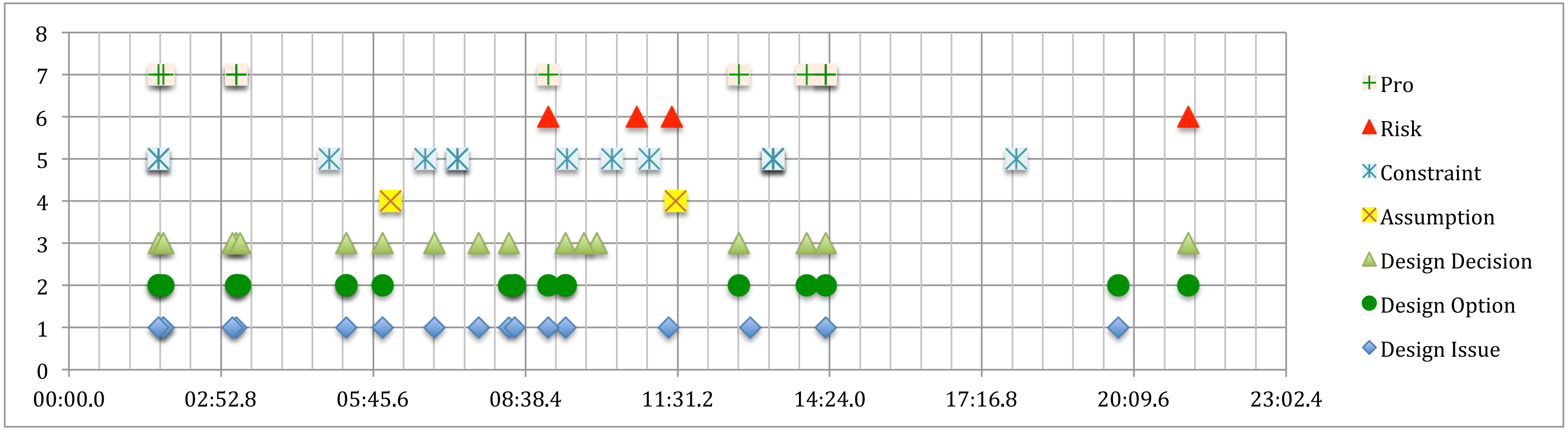
Time plot of design reasoning techniques and Design Rationale elements for Group 4



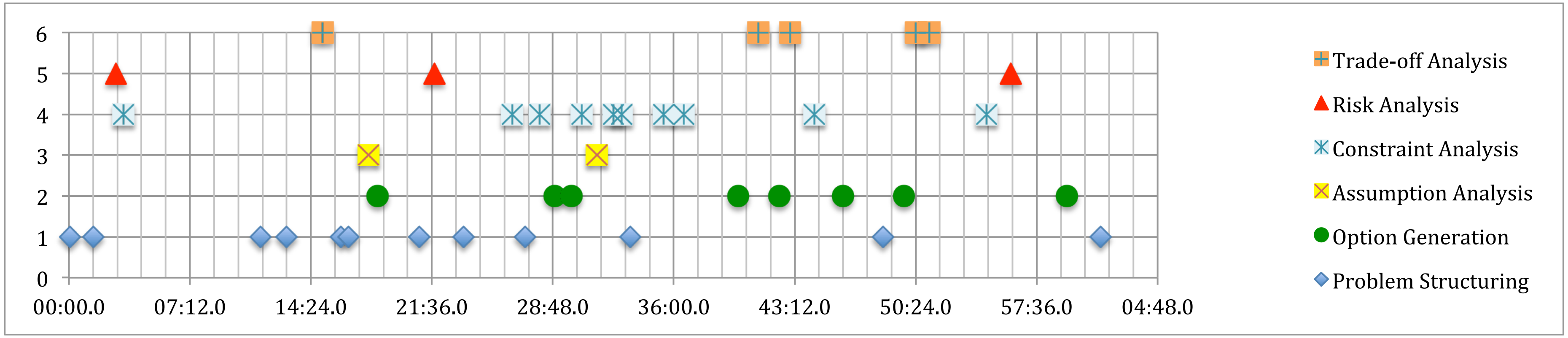


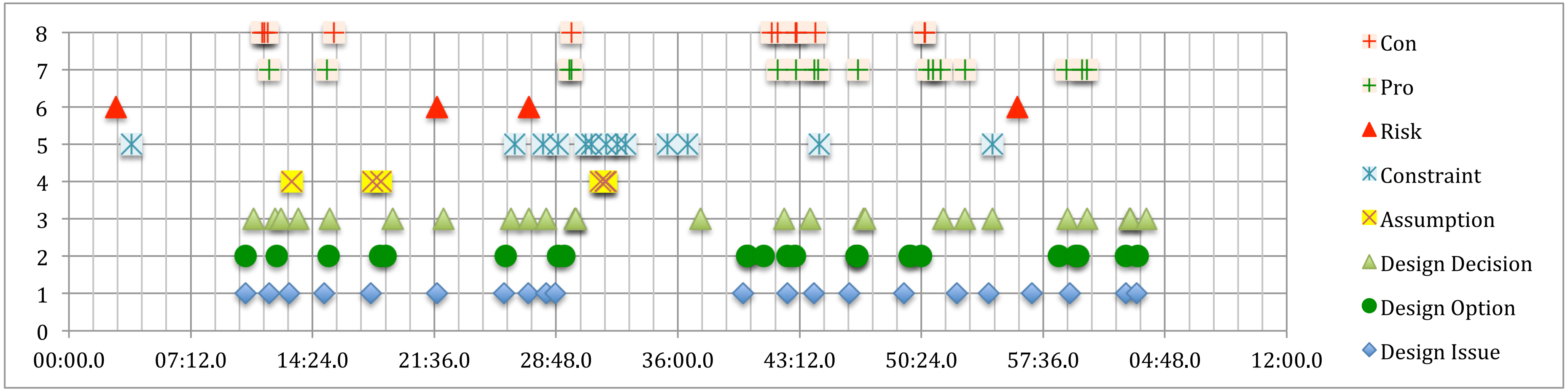
Time plot of design reasoning techniques and Design Rationale elements for Group 5


5
4
3
2
1
0
00:00.0
02:52.8
05:45.6
08:38.4
11:31.2
14:24.0
17:16.8
20:09.6
23:02.4
Risk Analysis
Constraint Analysis
Assumption Analysis
Option Generation
Problem Structuring

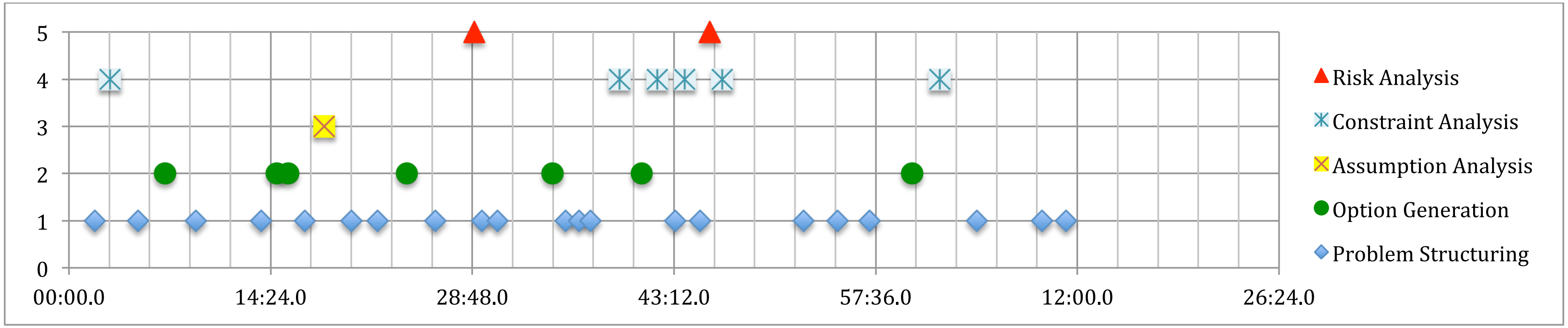

8
7
6
5
4
3
2
1
0
00:00.0
02:52.8
05:45.6
08:38.4
11:31.2
14:24.0
17:16.8
20:09.6
23:02.4
Pro
Risk
Constraint
Assumption
Design Decision
Design Option
Design Issue

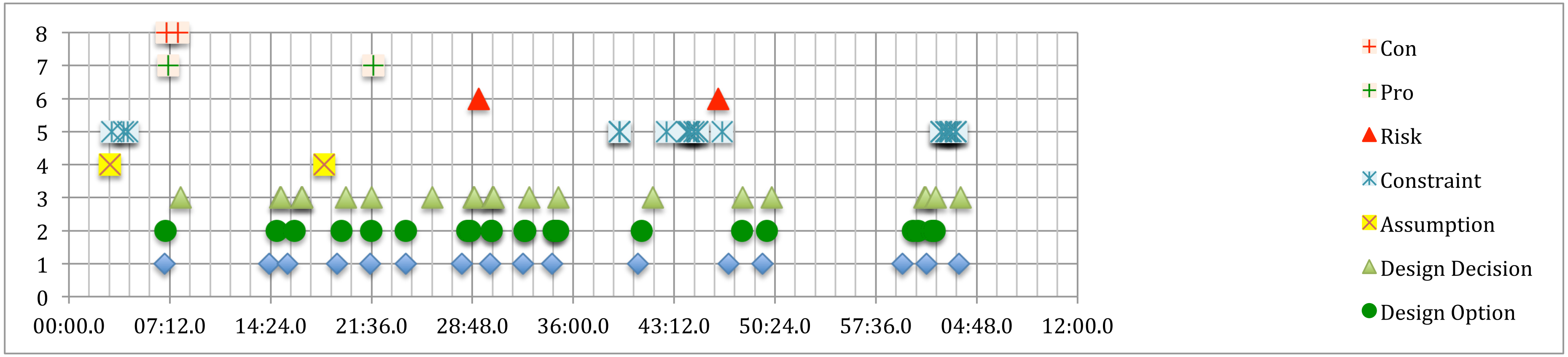
Time plot of design reasoning techniques and Design Rationale elements for Group 6



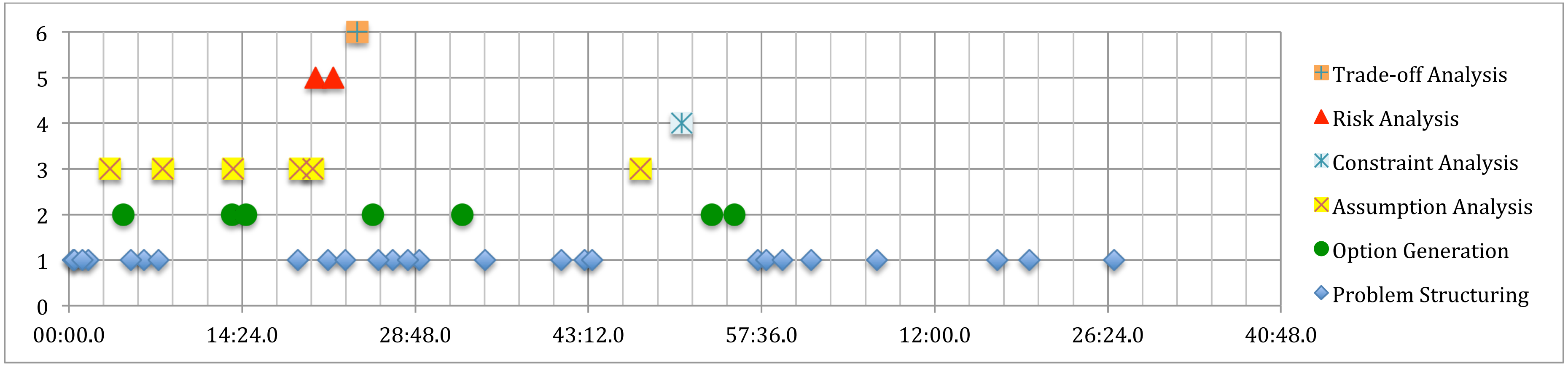


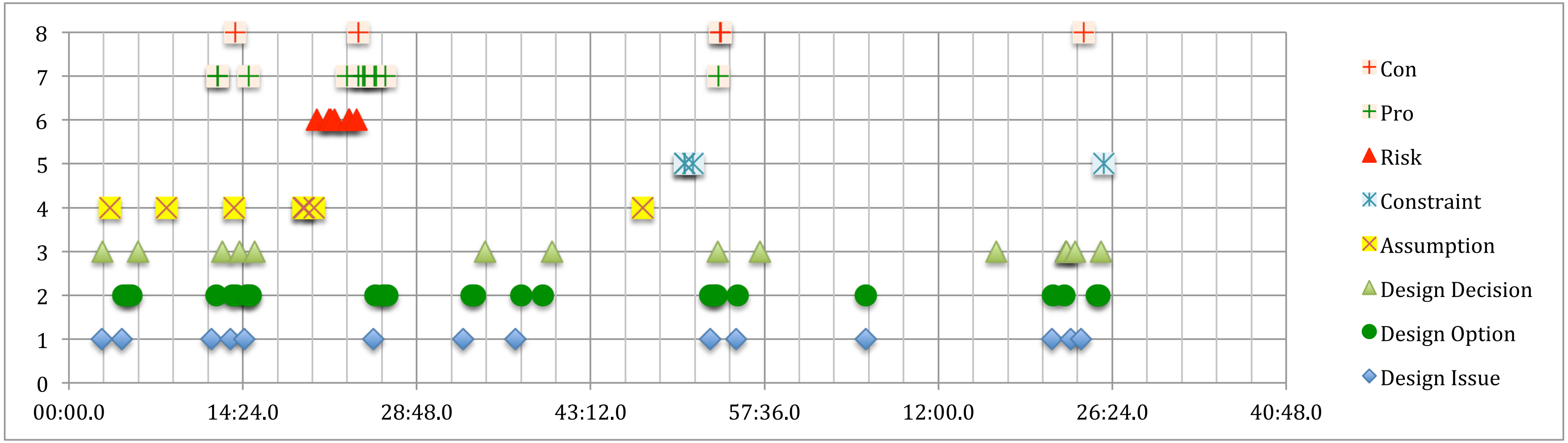
Time plot of design reasoning techniques and Design Rationale elements for Group 7



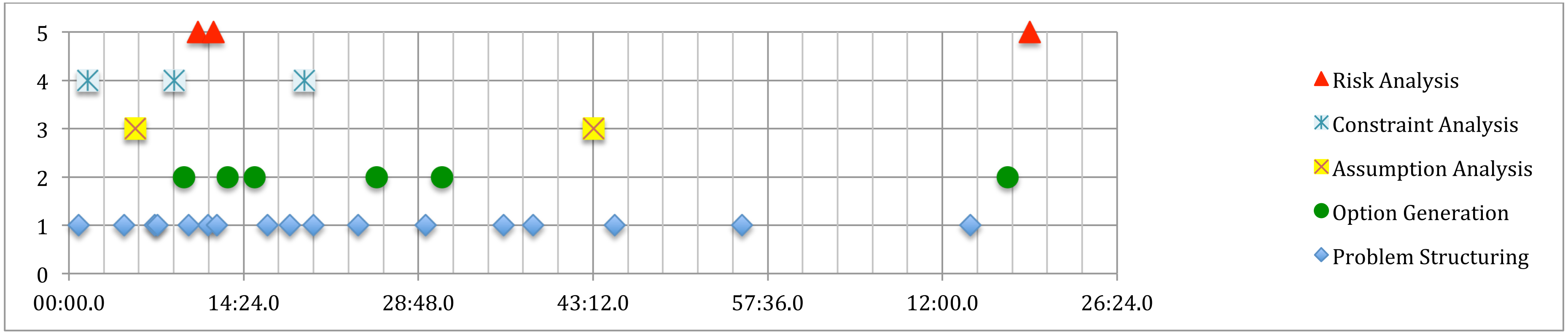


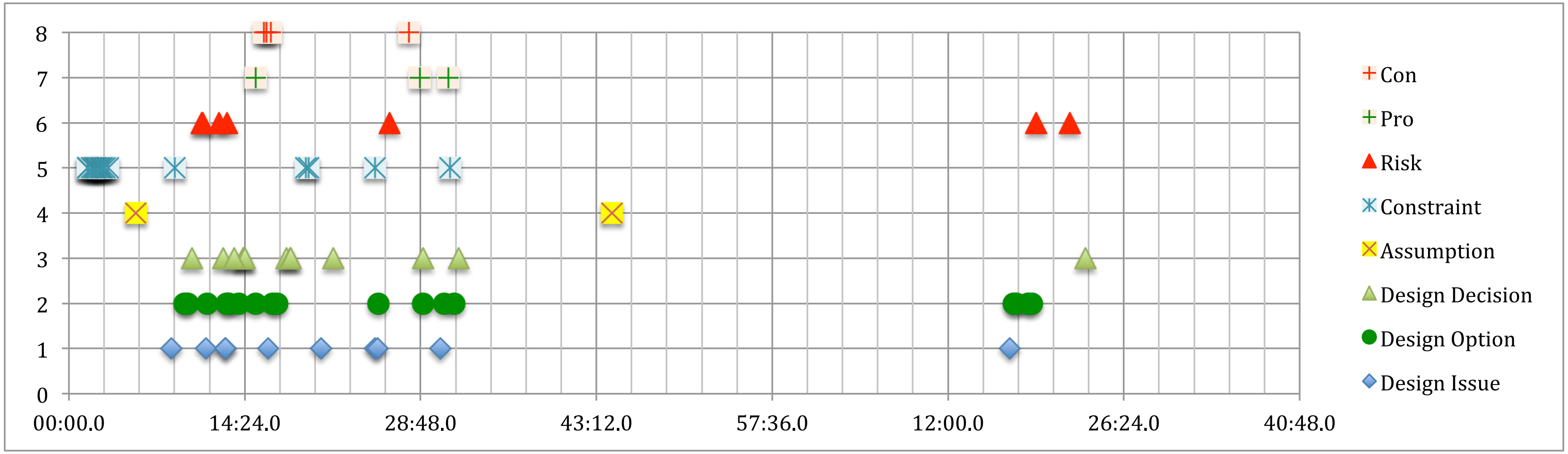
Time plot of design reasoning techniques and Design Rationale elements for Group 8



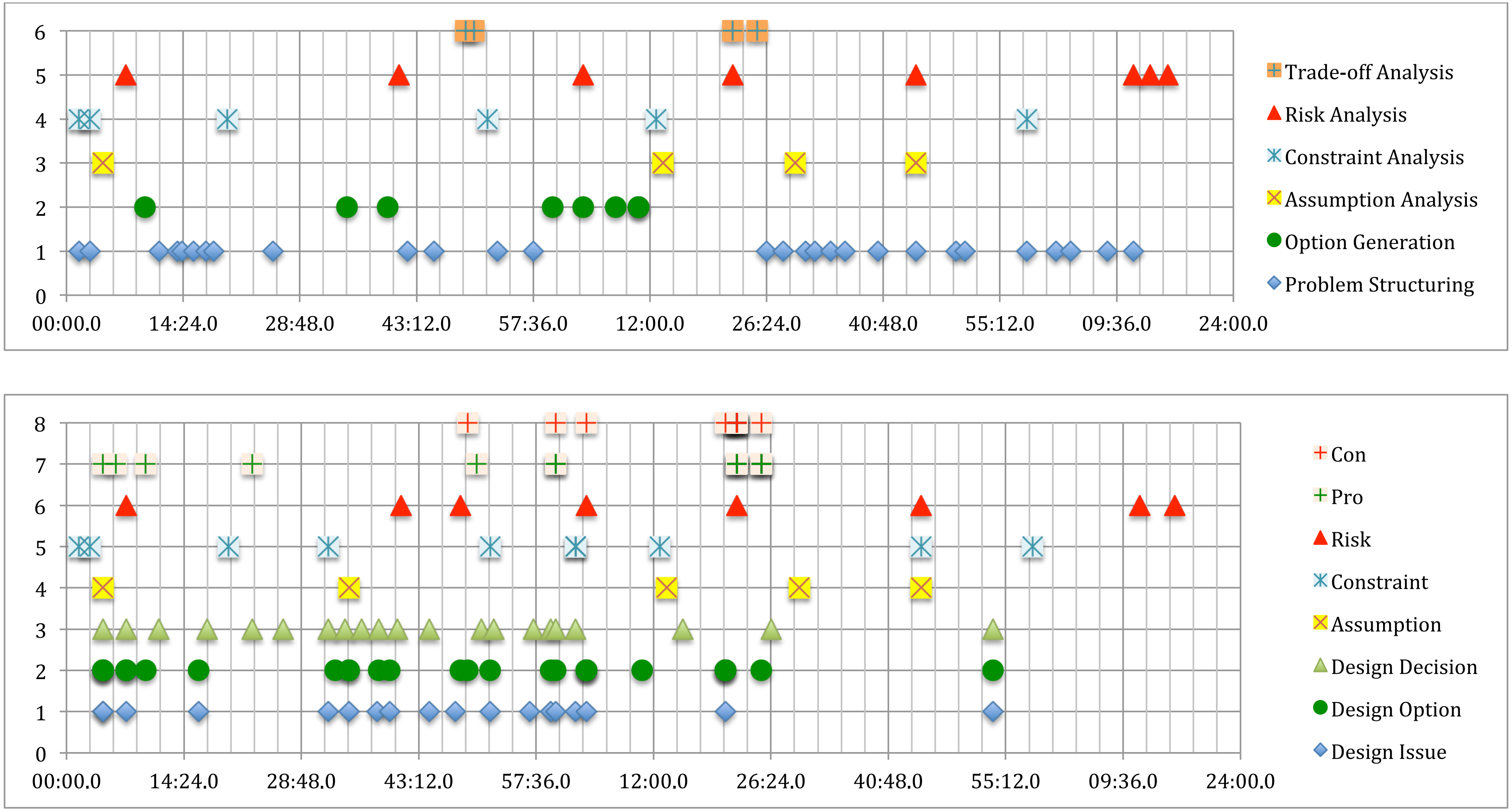


Time plot of design reasoning techniques and Design Rationale elements for Group 9

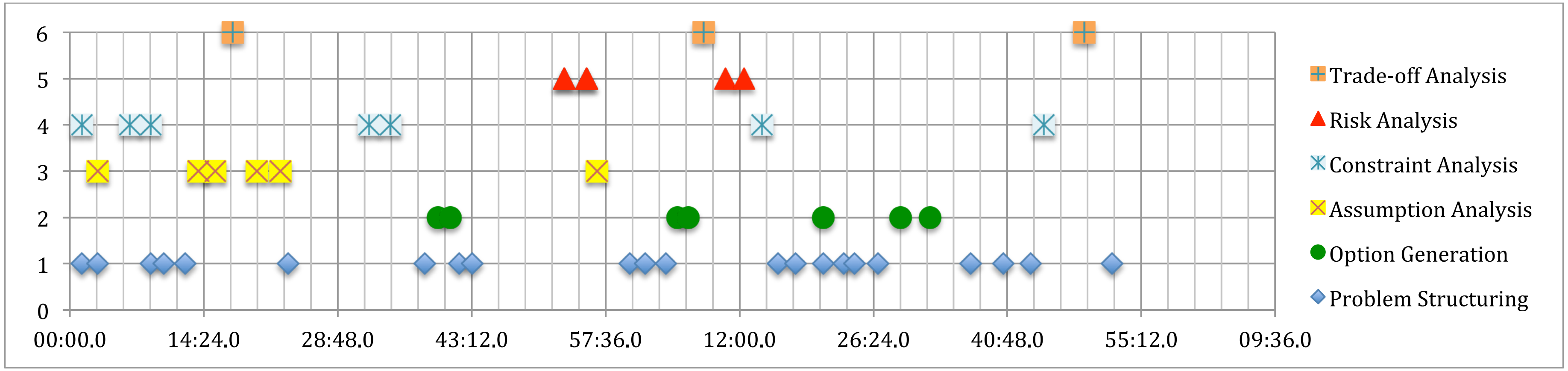


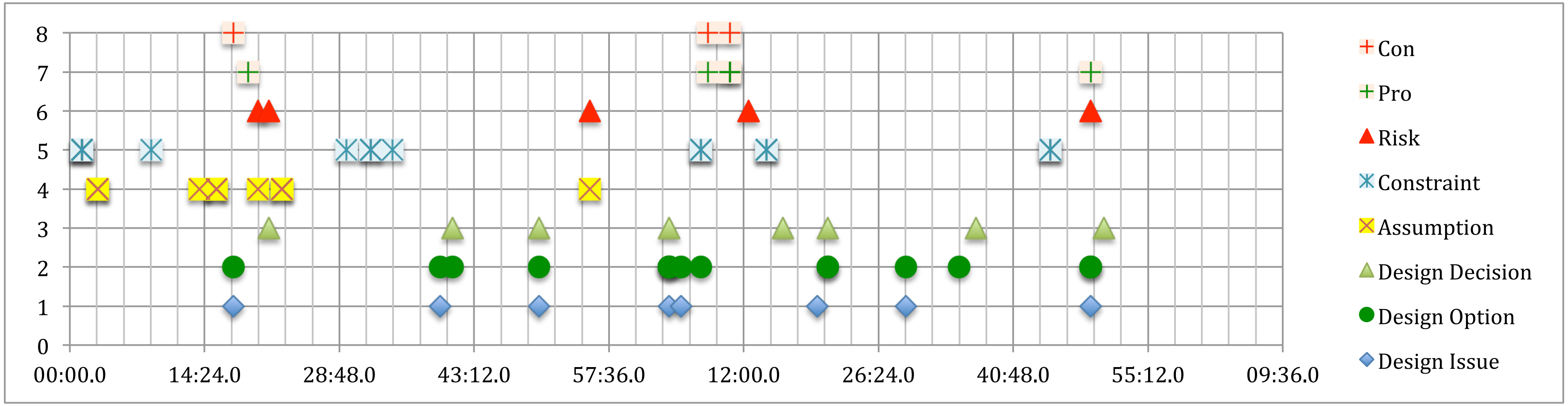


Time plot of design reasoning techniques and Design Rationale elements for Group 10

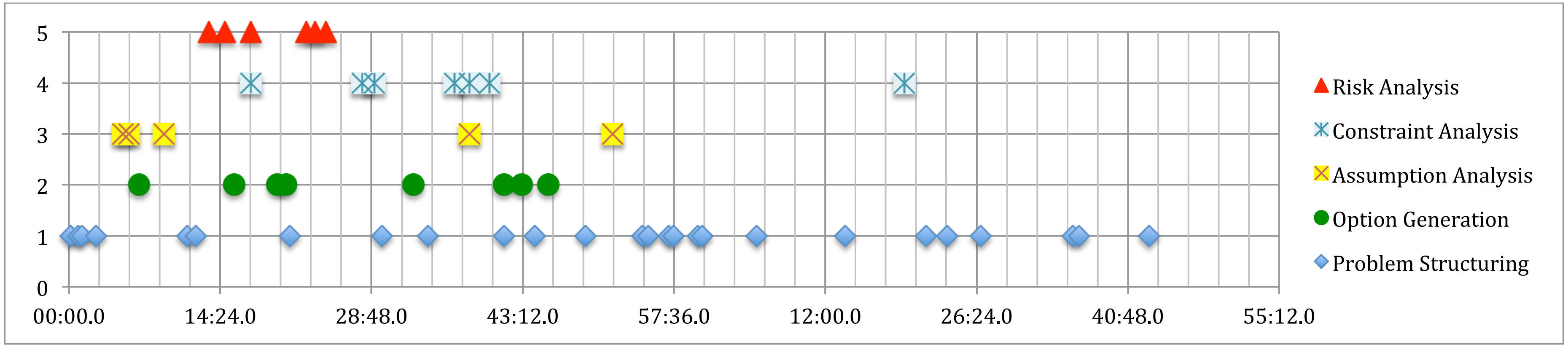


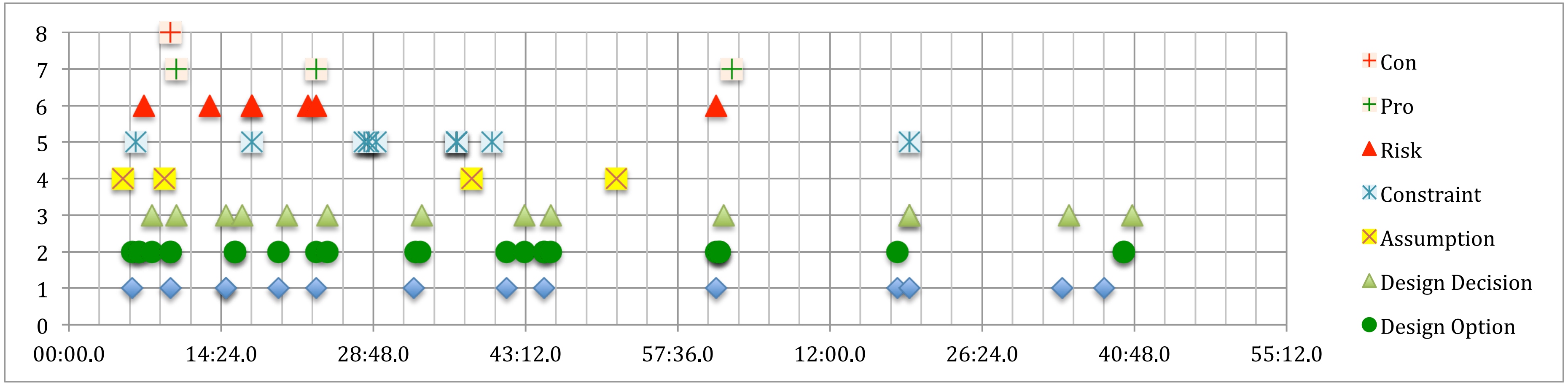
Time plot of design reasoning techniques and Design Rationale elements for Group 11





Time plot of design reasoning techniques and Design Rationale elements for Group 12


5
4
3
2
1
0
00:00.0
14:24.0
28:48.0
43:12.0
57:36.0
12:00.0
26:24.0
40:48.0
55:12.0
Risk Analysis
Constraint Analysis
Assumption Analysis
Option Generation
Problem Structuring


8
7
6
5
4
3
2
1
0
00:00.0
14:24.0
28:48.0
43:12.0
57:36.0
12:00.0
26:24.0
40:48.0
55:12.0
Con
Pro
Risk
Constraint
Assumption
Design Decision
Design Option

APPENDIX C

Event-Listing Matrix

Team 1

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			So one car can go left, right and goes straight forward And cannot change- pass to-	A	None	
2			The application is just a learning tool	A	None	
3	Are we going to use external software packages	Plugins	you can use a mathematical software program	Pro	Positive	External Software Plugins
			Calculate all the sequencing	Pro	Positive	
		Algoritms	We don't know what we're gonna use	Con	Negative	
4	How much traffic can one road handle		Students need to be able to configure the busyness of the roads.	C	None	every lane has a capacity.
5	Yeah if one car wants to go straight and one car in front of the car wants to go right or left, and must wait for a signal to go on green lights. Then the car that wants to go to the next traffic light must wait for the car in front of him.	you should also be able to configure on that crossroad, You should be able to configure the type of each lane, at minimum you need three lanes				You can have, if you have one lane, you can have one traffic light for all
		You can have, if you have one lane, you can have one traffic light for all				
6	Must work with sensors or without sensors, that detect-	I think these all fall under the bigger function, which is crossroad behaviour, or crossroad configuration				You should just give them the sandbox and they should be able to play, so that's one functionality.
		Can choose whether or not they have car sensors or not and that should in turn- or act behaviour of the traffic lights.				
7	a student cannot say, this traffic light straight on is green, while the opposite side left is green. Because that would result in crashes.		combinations of individual signals that would result in crashes should not be allowed	C	None	So I think we should limit the light behaviour and they should definitely be able to play with the timing
			That would complicate the system	R	None	
8	The students want. What kind of interface	be very abstract in our solution. Because we don't really know enough	Determine how the student wants to control the system.	A	Positive	we going to specify the requirements more by ourselves
		We should define the system for the architects to create it.	Because we don't know	R	Positive	
9	the students should be able to control the traffic density themselves. But we will have to determine how they're gonna do that.	Slider				you can simply say, every lane has a maximum capacity
		you can simply say, every lane has a maximum capacity	But that would also complicate the design	R	Negative	

10	So this is, wants to turn this way. But the light here is also on green so there will be a crash here with two cars	we can make it so that on the intersections they have little signs that say, no U-turns.	U-turns should maybe be illegal per definition	C	Positive	You can just say to the developers, just program it so that all traffic light combinations that collide are illegal.
			Which would simplify the mechanics	Pro	Positive	
			Yeah but is that going to be a question	Con	Negative	
			I don't really understand	Con	Negative	
		But this can also be solved with, you need traffic lights	But I think this is not really important now, because it's a very implementation detail.	Pro	Negative	
11	Are we going to set the busyness per road	Well if you have a traffic flow, you can start in the left area and come- might move to the right				Well maybe you could say the only variable for a road is the size so how much lanes it is, and that also determines the amount of cars that spawn on it, depending on the time of day
		for the entire area it's, for example, high noon and everyone's leaving for work, so the busyness is high.				
12	Also a road has a length, you can have different road lengths. Or do we want to make the road in sections.	Sections.	That would make it easy for the programmers to create the program.	Pro	Positive	so we're going with the tile based approach.
			So you basically drag and drop the map. I think this would create a very intuitive way	Pro	Positive	
			It makes sense. For the developers and for user	Pro	Positive	
			I think if we choose and approach like this it really defines the way you work with the objects	Pro	Positive	
			And also in the information view all these objects are tiles. And just in an intersection it would have traffic lights, and maybe you can zoom in on the intersection-	Pro	Positive	
			so, the point is that you can see, there are two intersections here and these are busy roads, or whatever, so you can see this road is an issue, and also these intersections are causing trouble. You can just zoom in and you can see the individual cars go to check where the problem is. Where the bottleneck is.	Pro	Positive	
13			And then we should make, have an assumption that we wanted it to be the most efficient. That we want the car behaviour to be the most efficient as possible, or maybe with random like in the real world	A	None	
14			It's a mathematical calculation, you can calculate and change the parameters of car length and waiting time and also the traffic rules. In the software	A	None	
15	But we don't know how we're going to visualize the architecture which I our main concern at this point.					I think we should have a module like the builder module, that allows you to place roads, intersections

16	I think you're going to want to adjust your timing scheme on a specific crossroad	Has its own timing scheme	Well I don't think they should be able to configure in such depth because that would really mess up the entire logic of all the traffic lights which I think is the point of the application	Con	Negative	It's separate timing scheme
			I don't think they should be able to override the existing algorithms	Con	Negative	
		There is an overall timing scheme				
17			For at least six times, in our case, because we have six	C	None	
18	we're gonna want to have a module to spawn cars?	randomly				I think every road should have, maybe separate road logic module, and if it's a road at the edge of the world it's a spawning road, or something.
		Just by a neighbourhood.				
19	For every spawner road you can configure how much car is spawned there. Is that an idea?	I think it would be best to have it only at sites.	So you can observe the flow a bit better I guess, because if you're gonna car spawn in the middle of your map then it's gonna be difficult to track the flow of traffic and see how your actions influence the flow.	Pro	Positive	so you only configure the busyness of the roads at the sides
20	I think we can cross out time of day, because we covered that with the road busyness.	Yeah. And students can choose to have, like, when you [inaudible] at night. And if it is not very busy they can say, I want orange light.	But then there will be a car crash. If two cars-	Con	Negative	Yeah let's leave it out for a minute
			That's their choice	Pro	Positive	
			That would be a feature of that night mode perhaps.	Pro	Positive	
			But then the rules change. Because cars can crash at night. That's the problem	Con	Negative	
			Cars are not allowed to crash	C	Negative	
		I think we should forget night mode for now				
21	Maybe we can set up the rules for how long the lights can go on green, yellow and red.	So, for example, the orange light is always like fewer than three seconds				Yeah, That's for the timing scheme
22	How does architecture look like, cause we have all this of the game and how are they connected.					The game manager can create, like, a blank area. Or can just load, like a, an area that you have done before. So basically the game manager just creates an instance to play in.
23	How do I fit the logic things in this?	I think that the logic is connected to the pieces you place.				Yeah, each intersection and each traffic light and I think some of the logics are unchangeable, but there are obviously settings which can be changed.
		Should we have like a logic manager module	The logics are all connected	Pro	Positive	
			If the students change one of the three logics then it can be a mess-	Pro	Positive	
24	how are we going to include the external software package					Shouldn't that just be part of the interaction logic? In combination with the timing scheme, that- which are already mid module. So we just say, that package is integrated into the interaction module

25	Like that graphics generation is that the proper way to model it?	With a line to the instance.	Yeah depends on how it's written, in which code I suppose	Pro	Positive	Module, straight to the instance
26	Do we need, like, and input manager or something. Like a mouse or controller	That's in the game manager right	It's really low level	Pro	Positive	Yeah that's game manager.
27	let's start to the information view	We were thinking to do a petri net	We have trade off, towards petri net, because I had some other work.	Pro	Positive	Yeah maybe we can zoom in on the logic manager, for example. And select the things the user would like to change.
			I think the information view is tricky	R	Negative	
			Because it's difficult	R	Negative	
		Well we could do our information view as an overlay on our functional model. Like, we could say just like FAM, We could select a scenario	But if you translate it back to the functional view you use all the elements in there. So it's not really convenient for us to use a graphical overlay	Con	Negative	
28	our next question is who does the instance and the elements.					The logic manager is the overall- well is the manager obviously. And it also passes things like borders between different tiles
29	How does this whole logic thing work is basically our problem.	Does the instance then send element list to the logic manager? Or is he gonna go- nah I think to the logic manager would then consult all these things.				Like, how do you call it, like a server-bus or something and then we or a broker and then it all gets put together
		Like, how do you call it, like a server-bus or something and then we or a broker and then it all gets put together	I imagined it like the user can have some options to change some logic.	A	Positive	
			The logic manager, shouldn't it communicate with the game manager rather than the instance.	A	Positive	
			I think this is the- this is not really an information view, I think.	R	Negative	
30	how do we do that in an information flow about the architecture How do we fit the builder and the logic manager in this process?	Sort like a database model	Yeah but that's not really an information view, I think it's more like a technical view	Con	Negative	A double block, so you just make an- information entities and the flows between them.
		maybe we can make a functional overlay				
		maybe we can make a UML with not just the class names, but also the information properties-	so you don't have a specific scenario	Pro	Positive	
31	How do you get the logic on an element					We build it, we create an instance with our element list, and then, it's like magic
32	Yeah. But now you want to draw the logic manager as well? Cause there's all the logic	I guess you could do this and just have the logic manager as the single box.	But then you have to define the information flow between them	Con	Negative	Yeah I think we should limit to this, that's maybe the best option for now. The entity we're making
		Like a sort of database.	much easier	Pro	Positive	
			In your model. Every road is connected to another road	C	Positive	
33	How do you propose to constraint? Because if you draw your first road, and you don't have an intersection	Not to begin with, but later on you can add some intersection				So it should not be a concrete constraint. Every road should have an intersection

34	but how to draw this like this. Then you have to draw also a line between road and intersection	But we also make like, our tile thing.	Yeah it's a bit difficult to	Con	Negative	I want to include with- I would say like and area as multiple tiles. [inaudible] consists of roads and intersections
			But is it not obsolete	Pro	Positive	
35	Find a solution for the sensors	Have traffic light sensors				I think that's a solution
		Have intersections	But they just affect the traffic light of that intersection. But it also affects the other traffic lights at an intersection, if you get what I mean. It doesn't just affect the traffic light in that lane it effects all-	Pro	Positive	
			The traffic lights in the intersection.	A	Positive	
36	users can select the size, how long it will be?	Select and then on the map and then, so long it gets				Everything is in the tile, but a road can have a type
		They add tiles. So you will add like, a road tile				

Team 2

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			the students must be able to create artificial map	C	None	
2			but the game engine isn't that the traffic simulator	A	None	
3			that's gonna complicate things a little bit more	R	None	
4	what about this stuff, the density you have to set it.	you wanna draw the map, you wanna set the timing, you wanna pass that data to game engine, and then you wanna set the density	so the traffic labels must be visual to the user in real time in the simulation	C	Positive	that look good, and those two will go into that, and that one will go back, and this one will trigger, the game engine will trigger the simulator.
			coz the game engine probably request some kind of random number simulator, and they get that from the packages	A	Positive	
5	isn't those two probably need to interact	yeah, if they gonna be able to put the traffic lights	but do you really need to know, do you really need to	Con	Negative	let connect this, alright
			coz it's hard to set traffic timing if you don't know if it's like 8 or 9 traffic lights	Pro	Positive	

6	if you don’t want to changes the map just the timings	you probably have to have some kind of arrow back to timing and map designer	ok and the map and the traffic is already save in the database. so can just choose that and then	Pro	Positive	ok, lets go for it.
		but then again if you wanna changes, you just restart, so you wouldn't have to like send anything from here from traffic, from the end, back to the start, you wouldn't have to send any data. you just	some thing like that,	Pro	Positive	
			I think that sound fair	Pro	Positive	
			but then what this arrow then, will is this necessary, because you never just send a map	Con	Negative	
		so you’re thinking like in the database you can have both the map saved and the map plus the traffic timing saved, ok then it make sense	I don’t know it could be a good	Pro	Positive	
			yeah, then it make sense	Pro	Positive	
			coz maybe it's even easier to just sometime have the map and then we do all the traffic timing instead of changing	Pro	Positive	
			I don’t think we need to have that connection , because it could be like confusing, I'm thinking about when we're doing the information view, then you would be really confuse about.	R	Negative	
			intersections light behavior should be able to change input from whether or not they choose to have sensors or not.	C	Negative	
7	I wondering if those two happens at the same time.	maybe we should have the one				so then I guess that is done
		called it simulates and visualizes game. coz it doesn’t make sense to first simulate it and then you show it .				

Team 3

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1	how can we model that		they say that you are only allowed to create your own maps	C	None	No when you compile the program, when you say one then it checks, ok, it's ok, or you can say, hey check my program. And then it tests it. Yeah I don't think it really matters when you do it. At least if you do it before it start running, the simulation
			that you have only four way crossings and no one ways	C	None	
			Does it have flash? I think so and you could do things	A	None	
			it says you can't have T intersections and stuff	C	None	
			there may not be one way roads, no T sections, and it must all be- always with the traffic lights	C	None	
			at least six of these intersections	C	None	
			but we don't know that because we can't look at the program	R	None	
			at least six intersections	C	None	
			All intersections will be four way	C	None	
			they are with or without sensors- or that you can determine yourself.	C	None	
			they must have traffic lights	C	None	
			intersections will be four way. No T intersections	C	None	
		is that a tree or something				
		that it's kind of a loop				
2	how do you draw	drag or drop, do it on a grid, click	Because if you choose drag and drop, you don't have to- like you don't have to put the possibility of a T- what is it called- T intersection on it. So you can just only drag a four way-	Pro	Positive	Yeah and if you already add the traffic lights to the crossings, then you also don't have that anymore.
			Yeah and it's way less complex	Pro	Positive	
			You also don't have those one way roads	Pro	Positive	

3	Do we have other rules. But they can have a varying length, so you should be able to drag and drop the lengths as well	If you have this cross then you can say, ok you can only move this. And like this, and you can pull it	Yeah you can do that	Pro	Positive	Yeah and then you don't have the problem
			maybe we have a rule then that you have, like, a minimum length. Because then you're not able to drag one way, like- so it's nowhere or- if you have this one, and you take this point and drag it to the middle then you have a T intersection	R	Negative	
			a crossing must have traffic lights, must be four way, each- how do they, each road has a length	C	Negative	
4	I mean, I think they mean that not all lights may be set to green at once	but I don't think that the crossings are linked together.	Yeah it doesn't really matter if another crossing is also in green, because then, it comes to the next one and if it's green then other traffic lights are not on green.	Con	Negative	so the crossings are linked together
		But then they should be, or the traffic lights are linked together	Yeah it doesn't really matter if another crossing is also in green, because then, it comes to the next one and if it's green then other traffic lights are not on green.	Pro	Positive	
5	But that is actually a problem, because we just said that each road		for the roads to be of varying lengths.	C	None	But then you can say that you'll have like some kind of margin, you have a kind of margin or padding between the, like the field you can draw in. and the actual point or places you can put your intersections.
			each road has a minimum length	A	None	
			But do we also need to make the maximum length, because otherwise you can do, instead of this, you can go the other way, you can still make the T section	R	None	
			the minimum will always still be there, you need the minimum, otherwise you cannot guarantee that there are no T	R	None	
			But then there are no right hand arrows.	A	None	
			No it says, also be able to- yeah but also on the- yeah I know what you mean because it doesn't say so, but because it says also left, it would mean- implicate like, also left, but also right.	A	None	
			then is it a risk that we don't know? The stakeholders, and therefor we're gonna do the wrong thing?	R	None	

6	we do not know who the stakeholders are					Ok, so we have the developers, the client and the software architect. And the students, the user
7			They'll be like, oh you didn't tell me about to do, oh I like this button here. Oh well, my program does not want me to put that button there, so now it's there.	R	None	
8	it should be possible to create a busy road, or a seldom used one. And any variation in between. How exactly this is declared by the user and depicted by the system is up to you.	Integers. Shall we till one hundred, how many cars per second.				Because maybe they have a much better solution for this. But as a user you also need to know which road you're going to increase.
		Yeah rush hour. I don't really know how else to do this. I don't even know if we have to say this or that we can leave this to the developers.	For this simulation thingy. Because, we are now designing the system and we also have the developers who need to be able to design the system. But if we give them this, how do they know what we really want from them. If they have it visual it might be better for them, more clear what they want	R	Negative	
		maybe we also have to define a minimum or maximum speed	We were not allowed to create dangerous things, and saying that one drives one hundred, and the other one drives 30	C	Negative	
		The input roads get numbered, and then for every input road you can say, ok, I want this density to be 10. Or 90.				
		Or you have a slide bar from low to high	Yeah because if you have a very busy road, the other intersections need to participate on that. Like, you have to get the cars away from wherever they are	Pro	Positive	
			Oh yeah we also have the adjustable options. Something has to happen with them	R	Positive	
9	Can you share your map	No, don't make it too difficult	Out of scope	Pro	Positive	So you only can save and open a file
10	Maybe we should be more specific communication with the system and the server of the program	We do have something like this in [previous project]. That they start a local server with a database to put the current program in that, the current map. So maybe that he meant that.				It sounds like information right
11			Ok, only the operational viewpoint, because there you say what kind of systems you need, like I said, windows or mac.	A	None	

Team 4

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1	we have to determine how we wanna save the data then I guess, because that is also relevant for the information viewpoint later on.	some sort of xml or something.	yeah. that would be fine. yeah that would be the best.	Pro	Positive	so xml that's a file system. so we gonna have a file system.
			I guess it also the easiest form to implement. if you are talking to what was it. students who just complete basic computer science degree.	Pro	Positive	
2	let start with the first one.		they must be able to create a visual map. they must describe behavior of the traffic lights. and simulate traffic flows, and change traffic density.	C	None	but maybe with this time constraints would be better to make a state diagram with the four main functionalities, instead four sequences.
		the message sequence chart.	sequence chart more useful for that but now we have to make it like four.	Con	Negative	
			I'm rethinking this is the interaction between the system and the users it really best way to show that in the state diagram, because we are now describing state of the system. they can change but they might not be change due the interaction. not really. doesn't really follow that well from the notation.	Con	Negative	
		the state diagram	in one state diagram we can cover like all those four functionalities.	Pro	Positive	
3	I guess this is the right way. so this is UML do we intentionally has it to do.		that covers like all the functionality basically, sort of.	Pro	None	UML sequence. I guess that's it.
			yeah we could do that yeah. because there is kind are some sequence.	Pro	None	
4	I like traffic light place automatically when there is basically an intersection created.		every intersection on the map want have traffic light	C	None	so yeah than it would be the best thing that when the road is place and then automatically it's basically made an intersection. if it's like overlap or something. and then immediately some traffic light are place
			four way basically means like just.	C	None	
			there is no T.	C	None	
5	it's look like in the object scenario it would also communicate with the MSP because when you create an object	on and off	after altering the map's timing scheme they should immediately see the changes in the traffic pattern.	C	Positive	real time response.
		always real time changes.	we can't make it too difficult. because we will use a lot of functionality.	R	Negative	

6	is the way do it like. creating a road or creating object.	creating a road				so we will create object.
		creating object.	maybe that's some kind of module. well seeing from the perspective of the students they can only create the road, its seems from the perspective more of the system. and the functional view is more like seeing from the systems right.	Pro	Positive	
			we might be thinking to much in the information view.	R	Negative	
7	do we add the multiplicity already now. or do we connect them with the those crowd feed. or do we connect them with some like in the UML class with some multiplicity. I believe you have to get association or yeah association is the right one.	so a map can have multiple road. and road can be in one map. and a road can be in as many intersection as it wants	intersection consist of four roads.	C	Positive	so we do basically state that you have intersection and they are connected by road. right. not road are form intersection by crossing. but also give them number and in the glossary we define them.
		remove road. intersection has a traffic light. I would not say it a road. so traffic light sensors connected to traffic light. it placed in the road.	each map at least accommodate six intersection	C	Positive	
8	there's a road between intersection but also there also road from which vehicle enter the map. so different kind a road. so we have to make another road.	map road. enter map road?	you should able to changes the traffic density that entice the map on different road	C	Positive	and this is entry road
			it not stated in the assignment but it should be because there's really connecting road.	Pro	Positive	
			but does it state it anywhere there should be such road.	Con	Negative	
			the T intersection shouldn't be allowed	C	Positive	
			how the user can change traffic scheme. but. I'm not sure you.	R	Negative	

Team 5

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1	how do you call it	but it should do that automatically	it should not be left to the users to specified that the opposite side should turn red. that should happen automatically . if this is going green then no matter what the opposite side should turn red because other wise it will be a crash.	Pro	Positive	so it's not an option it will be and AND condition.
			the constraint is to create red one on the other side.	C	Negative	
		shouldn't be left to the user I mean if the user selects that left turn	because there is no one way road.	C	Positive	
2	how can we include that. opportunity for the student that he can decide. if they want to go right or left can we model this in some way.	well if they want go left. they should switch to the left lane of the road and the sensor should detect it	so then if you have sufficient sensor trigger that so many people are waiting on the left lane. then the left signal comes on after sometime. but that can be choice for the student it could display that. you know you have three cars waiting in the left lane so do you wanna turn the left signal on now. have it like that.	Pro	Positive	okay. that could be.
			it says no one way road.	C	Negative	
			but when you create this a database would that table street it is possible there is no street in it.	R	Negative	
			T is not allowed. only for.	C	Positive	
3	but. then we should model the opportunity that the traffic light is on one side or on the other side. how to model this.		I think someone should create this system so should be really a plug in or should be already created because we can not create such a system because every traffic like information should be in that simulation how to get that information.	R	None	yeah. we all have to connect this that's the problem now
			like you know anytime you say you want to see you running the program and you want to see the flow of traffic immediately you have to get the data from the satellite other wise you can't see anything.	A	None	
		oh maybe a lane has a traffic light.	that would make it easier. because see. we are considering this whole street I say Oudegrah street this whole thing. both side of it . okay and up side will have traffic light down side will have traffic light.	Pro	Positive	
4			busy road or seldom use road or anything in between.	C	None	
5			yeah. because when you have a street and car wants to turn left you have to cross the other side of the street and that's. yeah. you have to make sure. that from the other direction no cars is coming.	C	None	
6			you need that left hand green arrow lights.	C	None	

7			left hand green arrow yes and at the same time the sequence will be that from the opposite direction should be red. those two should be match together.	C	None	
8	how to align to understand the function now. functionality..	the decision model get state back from data source and creating module	yeah. creating street view or something. which will be more like a database I guess once he has created it will be like static database. he is not gonna update the street every time.	Pro	Positive	but the this decision part will be dynamic depending on the of the date the flow of traffic
9	but now we have still the problem how to connect this. we have intersection intersection has a . vertical street and horizontal street.	well we can connected by having a traffic light being associated with an intersection	and then the traffic light becomes like an independent entity. we can place it anywhere. in the map and then we just connect this property to it. as the property of traffic light.	Pro	Positive	that's good idea. so intersection is a property of traffic light also.
			I think that would be simplier than trying to put import traffic light into each of this.	Pro	Positive	
			all the intersection are four way intersection	C	Positive	
10	but I'm thinking about how to model this because you have to this is like a method or something because you have to hold you have to. I don't know how to say because yeah you need other way to model this. because a street has a right or left side always.	but then we have to make two classes out of it.				so this traffic light will belong to Z intersection oudegrah street left side. so it has all information about this street the sides of this street and the intersection.
		can't it just be an attribute out of it. which can't be null .	we don't know how to refer this attribute	R	Negative	
11	you mean how we measure it.					cars per time
12	I don't know how to put this.	as a default property that each street has three lanes.	but we only have that option four cross or four ways intersections.	Pro	Positive	yes. we can just specify like that.
		or should we let this to the students that he decides if there should be third lane or is two lanes are efficient. because. that's I think also. we can make it Boolean. and then we can just put our own notation for that saying zero means is going.	yeah we should definitely have that option because not all street are gonna be so broad street even if it is a four way intersection it could be a small intersection you know.	Pro	Positive	
			they should have the option to disable the sensor	C	None	
13	but now we also have that intersection problem.	so has a street intersection				intersection is property of the map, easier to handle.
		the map a street and the intersection.				
14	so basically when they are creating they should be able to see on the map how like we have should have option for showing on the map visually the traffic or whatever what is going	we don't need to show individual car but you know like Google map shows like the whole road becomes like red or something if it's caught like a lot of traffic				so we can use that maybe.
			when ever we change the traffic light or the signal we will not be creating new street.	A	Negative	
			road of varying length.	C	Positive	
15	how do we specify that none of this can be null.					okay just put a bracket as not null the developer will.
16			you approach should readily accommodate at least six intersection	C	None	

17			constraint that student must create six intersection at least.	C	None	
18	this just to see it just so the user can see it on his console that okay .					so we have to model or we have to give the opportunity to the students to play and to stop the simulation.
19	update	so the update should get the data from some satellite right.				yeah. or when user make some changes the simulation should stop and then make the update and then start again.so this is also function of that.
20	yeah okay but we need to specify where the traffic light is. exactly on that street.	by length we can do that by length of course.				yeah fine if it's only the student that are using it then maybe we don't need it
		of course from the starting point and how much distance.				
21	how to include this in our application.	we'll just keep it as external data sources and line if they want to refer to more.	well it will see that at this time this road has this much traffic. which information the traffic light cannot give you. you know. you can see for the whole area or something or each traffic light will have the information of how many cars are only passing this traffic light. but with the GPS you can see with the satellite thing you can see how crowded everything is.	Pro	Positive	yeah including. yeah that is good
			does it changes to the system	R	Negative	
22	the option.	yeah. whether you want to take input only from the satellite				maybe we should leave that option.
		do you want take input from the previous traffic signal also.				
23			the sensor we should be able to sometimes like use or disable or however they want	C	None	

Team 6

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			If we make a wrong software architecture, the software is going to get build wrong	R	None	
2			all intersections will be four way. There are no T intersections nor one-ways ok. Must be able to design each intersection	C	None	
3	which was relevant here	if you wish you may assume that you would be able to use an existing software package that provides relevant mathematical functionalities such as statistical distributions, random number	why would we need to	Con	Negative	I would go with queuing
			Why would we need statistical distributions or queuing-	Con	Negative	
			traffic lights not- we don't have to do anything with the speed	Con	Negative	
4	you still have to create the road, the density of the cars, the number of cars, their speed, the left turns	did you want to combine queuing theory with mathematical as well.	So I think mathematical functions here are needed	Pro	Positive	Let's then create one entity and we call it mathematical functionality
5	when you model the environment we then have to explain each of the components that we drew		this software program, it's going to be created not for the civil engineer student	A	None	so I think environment is just, UCI course
6	Do you show interaction between a user and the system in the context, do you, already? Or do you just show the high level overview	just with arrow	Yeah, you do show the relationship between the system and the user.	Pro	Positive	Ok. Just draw, like, an arrow that says
			Then you're going more to use cases. And a model because traffic system isn't one functionality. I mean there's one package-	Con	Negative	

7	I think it should be because you need to place cars		I was under the assumption that they were only busy with roads and traffic lights, and not necessarily cars as entities.	A	None	My pic would be, go with the car instead of guessing that the program would do that
			Does it? I was under the assumption that there was only changing like, traffic	A	None	
		It doesn't specify if one of the systems does that for you automatically, or if you				
		I think the easiest way to go would be creating the car				
8	it's gonna be really difficult to specify an entire process of the other ones that you have		because there's not enough information. A lot of this is going to be under the assumption	R	None	So you specify the map and then the second part would be, you specify the pipe of the road
9	What kind of patterns though. Would you be able to select	a road pattern	The resulting map need not to be complex but should allow for roads to vary in length, to be placed in different arrangements of intersections to be created	C	Positive	Ok, so select a road pattern, then we agree on that one
10	What are the types of intersection		We don't know	R	None	but it doesn't matter because we're just modelling the process. So that could be in petri nets, just one or the other. Going backwards and forwards
			Your approach should readily accommodate at least six intersections, if not more	C	None	
11	How do we model restrictions in FAM?					Was it with QA notation

12	Students must be able to describe the behavior of the traffic light at each of the intersection	So that would be a sub process of intersection arrangement. That is not a separate step but it's- once you select	for each of the intersections you can have a minimum of six, and up to infinite, for each of them you need to specify the traffic light	C	Positive	The traffic light would automatically be there as an intersection, but the behavior You specify later
		Can't it be a different step altogether	because you're setting up these traffic light after you select the intersection	Pro	Positive	
			But it might be better to do it afterwards	Pro	Positive	
			but in essence you could have them select six intersections, as it would give you the option to model them	Con	Negative	
			Your approach should also be able to accommodate left hand turns, protected by left hand	C	None	
			Should be able to accommodate left hand turns, protected by left hand green arrow lights.	C	None	
			But that's in sequences already defined right?	A	None	
			We assume this one is defined in the traffic lights-	A	None	
			I think we should assume that this is done in sequences and timing schemes	A	None	
			Combinations of individual signals that would result in crashes should not be allowed	C	None	
			it should not allow for crashes	C	None	
			every intersection of the map must have traffic lights, there are not any stop signs, overpasses, or other variations. All intersection will four way, there are no T intersections and nor one way road	C	None	
			Students must be able to design each intersection with or without the option to have sensors that detect whether any cars are present in a given lane	C	None	
			Your approach should readily accommodate at least six intersections if not more	C	None	
13	it would display it. Start the simulation and then it would display it in real-time	Window				Gives two more options, which is media player functionality and exporting function.
		frame				
		Have exporting	from my point of view that's not really viable in terms of the software that they're trying to build because if you want somebody to learn-see all the interactions, they will want to play it instantly. They will try to model things and then play, I want to see it. That's one thing and as the second thing is, that's a requirement of the system	Con	Negative	
			The export option I think would come in handy in real world	Pro	Positive	
			It has to be presented in real-time to the user. To simulate traffic flow on the map, so we need some sort of player	Con	Negative	

14	it's up to you how to represent this information to the student using your program.	you may choose to depict individual cars or to use a more abstract representation.	HARDER	CON	Negative	we pick the individual cars representation.
		Just visualize all the cars	It's easiest	Pro	Positive	
			The current state of the intersection traffic lights should also be depicted visually and updated when they change	Con	Negative	
15	But why individual cars representation?		Cause it gives you more accurate information	Pro	None	we pick the individual cars repr
			Well I don't know, what would be the higher abstraction of the-	Con	None	
			I can't think of anything that's better than individual cars in software packages like this.	Pro	None	
			Students should be able to change the traffic density that enter the map on a given road	C	None	
16	they should be able to change the traffic density that enters the map on a given road	Specify the road characteristics	HARDER	CON	Negative	So we just give them- just gonna enter an integer, the amount of number of cars-
		Just like a number, Like an integer	I guess that's the easiest	Pro	Positive	
17	Type of cars, because you could have trucks, you could have personal cars	Does it calculate the size of cars	But then you have- you need to know the length	Con	Negative	it's not on the side of the system. That won't be the logic behind it because, logically speaking, if you have to select something you're not gonna care about how much that selection actually weighs or something, you just need the selection
			and you need the seize for traffic digestion yeah	Con	Negative	
		I think it's best to specify on, not specify on a motorcycle or car or truck, but on rate.	Well we can enter that into the system. The system knows, like an average length of a car or truck	Pro	Positive	
			So the user doesn't need to know about that	Pro	Positive	
			Let's make- that process is easier to include then another process of defining what is a car and how big is the car and how big is the truck.	Pro	Positive	
18	from usage perspective, doing the activity, they don't care about the weight, they just care about selection. But when it comes to the system, that would need to be modelled, that- of course. But those are basically physics, those are, I mean-		vehicle spec should adhere to like, gravitational laws, the laws of mass	Pro	None	Vehicles specification, I think that tells enough

19	If you just want to see a road visualization simulation		That might also be a restriction that we, kind of come up with is the fact that the user should not be limited to specifying all the characteristics of the simulation	C	None	The simulation should have a default option
			Yeah but my concern is that, when you have, for example, you specify a change timing, and you do the visualization. Visualization runs, it doesn't matter what you specify. And everything can crack, for example, although we've specified it's not allowed, but in some cases it might because of some strange, I don't know, combination of vehicles or whatever. And they should be able to see potential problems on this visualization, so there could be like a window, for potential problems, and it could just be intersection, six, I don't know, the timing is incorrect. Or something. It's like a warning	R	None	
20	the visualization should, I guess, support the option of going backwards and altering it	Create something, simulate it and then go back and change it. Change it and visualize it	So you can edit it directly when it's wrong.	Pro	Positive	Yeah basically yeah, so this would all, all of this, I guess, would be in a graphical user interface as well
21	there should always be a link or trajection to a process that's always checking if it's correctly or not	do you want a validator that's validating at the end of view				So, a validator, I guess, on every single step
		Or do you want to validate that- validate after each step you're doing. So creating an intersection, changing the speed and the timing on the traffic lights	if we have a lot of inputs for like, the mass and speed	Pro	Positive	
			then you have a problem solve validator would be good. Also this would help with all the constraints that we have, with like, the different intersections- you could immediately check if the intersection was a proper cross, or was it a T, was it this, was it different pattern, was it something else.	Pro	Positive	
22	But you still need to get the laws into the system before you can validate them	It would just get the data from different external entities-				Validator functionality is internal
23	How do you model it	Internal				Validator is part of the TS, the TS gets all the data-

Team 7

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			The resulting map needs not be complex	A	None	
2			so it should accommodate at least six intersections ans also of varying length.	C	None	
3			not a T. So only- and also not one way roads.	C	None	
4	What notation are we gonna use	petri net	Yeah ok but a petri net is just one process	Con	Negative	I think this one is at- too functional. Maybe we can start with context and then information
			you can take decisions into a petri net	Pro	Positive	
			But is that information flow	Con	Negative	
5	I don't know how to add this professor into the context view.	So maybe, for example, the professor can create, redefine it's scenarios and the students can rely on it for testing their own				Maybe you have just, with professors and not professor E
6	The scope	like the static view and dynamic view				Yeah we can elaborate on this
7			I think we can still do developers here. To the system	A	None	
8	Should have a link with an outsource program for the statistical distribution	So let's go with just software, existing software package.				Just a software package ok.
9	What notation did we use actually	UML	Yeah. So then we just have to change processes	Pro	Positive	I think we should do UML
10	Look maybe we can do it like this	global				A bit global then, a bit abstract. And the output is a simulation.
11	how do you model that	real-time visualization				So it can be like a report. Yeah, or maybe if you put simulation here, that flows from static and dynamic, you can redirect it to the outcome or something.
		Document	I think it's not clear enough	R	Negative	
12	What is to be communicated between different	OR				It's for the data, it's an OR And for the system it's an AND
		AND				
13	but I don't know how you're gonna put in the draw?	travel rules				only for dynamic
		Traffic information and sensor information				
14	how are you gonna see this	traffic view				traffic timing scheme
		traffic simulation view				
		traffic timing scheme				

15	We should include in which view, like, these automatic constraints		they should allow for the roads of varying length	C	None	I think it's a module that's going to be activated after you've designed the map
			know how to- different intersections, and at least six.	C	None	
			if one traffic light is green, and the other turns green as well, but a crash could happen. That cannot be the case	C	None	
			so six intersections	C	None	
			they have to be four way	C	None	
			And every intersection has to have traffic lights.	C	None	
			Combination of individual signals, cannot lead to crashes	C	None	
			Combination of signals	C	None	
			four way-	C	None	
			otherwise you can't change, because when you don't know what's	R	None	
		it would be a module				
16	How can we implement that?	density checker	students must be able to change the traffic density that enters the map of a given road	C	Positive	Something like that yeah
17	So now we are going to separate the functionalities of the simulation, functionalities of system global functionalities, or- and the rules of the system?	No. well, I don't know yet how we're going to incorporate this and this. Because this is- But we can make this a model, like for example, rules management or something.				Yeah, that's good. And then just mention in the functional view, rules management.
18	what about light behaviour, we can just, yeah, maybe we have to specify what we want to know and that's the current state.	Light visualization or something like that				it's pattern to constraint
		You need a pattern				
19	Maybe we have to check how many cars are- we have to set a maximum.	It will be an outcome of the process, like, you want to give like, how many cars it's supporting on this intersection, for example.				we make it a rule
		Do we make it a rule	At least six intersections	C	Positive	
			Maximum of car	C	Positive	
			Maximum of waiting time	C	Positive	
			Minimum speed?	C	Positive	
			Every intersection has to have traffic lights	C	Positive	
			And also has to have the four way	C	Positive	
			There's to be a four way street	C	Positive	
			And the combination of signals cannot lead to crashes	C	Positive	
20	I don't know how we can put this, like a model yet					Just maximize the view on this

Team 8

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1	what to name this then					Modifier
2			Existing software package	A	None	
3	that should be cross platform right. Then we also should model like it should run on Java	Yeah, yeah maybe, can't that one instead of queuing system, just yeah, statistical package or statistical- cause there are more things that can be supported in the simulation right				So one is for windows, one for macOSX and one for linux
		And maybe the model, the operating system, package everyone. For example, one windows, package one for macOSX, and one for unix based				
4			So that's more than just, yeah, just simulation package. If you call it like that then we are, yeah, of course already assuming that a lot of package is used, but I think that's, yeah, an assumption	A	None	
5	Yeah I was thinking is it a computer application, or a web-based application. It doesn't really say-	web application	I think it's more accessible	Pro	Positive	I would say then we also have a web application
			Multiple platforms	Pro	Positive	
			It might be easier to develop	Pro	Positive	
			Yeah, it's a risk that you need internet	R	Negative	
			yeah of course you want it to be dependent on a separate package of course, this [inaudible] package is a separate package. By another vendor of course	R	Negative	
			You have to be online	R	Negative	
			I think we have to make an assumption that every browser renders the output the same.	A	Positive	
			assumption that it is a web-based application	A	Positive	
			We made another assumption that a simulation package is used of course	A	Positive	
			Assumptions could of course be a HTML 5 right now	A	Positive	
			a risk might be of course that- of course there is a so while you are travelling. For example, when you have an older device that could be a problem of course. So then you couldn't use the navigation	R	Negative	
			that might be a risk of course by doing it in the tablet it might be slow. It might be slow on a smartphone	R	Negative	
			need enough computing capabilities in your device to run the simulation at all	R	Negative	
			consider risk to depend on an external company to provide functionality	R	Negative	
			doing it through internet, having it supported.	Pro	Positive	
			But it might be slow on the tablet	Con	Negative	
			You can also maybe program it yourself then right?	Pro	Positive	
			maybe focus on some open source depository somewhere	Pro	Positive	

6	I'm still thinking about that left hand turn. We didn't include it anywhere. How can you model it	it´s actually a property of your road	Should we really go to such a deep level then	Con	Negative	Just can name them that it has property road then set that in the paint properties
		split it between parameters and rules				
7	how does it simulate what a single car will do	That is not explained here, it´s just part of the black box of simulation				Yeah we can have that
		maybe we can add in the simulation box, like, randomized car movement, something like that	You cannot decide for every car where to go for example, not in this basic simulation	Pro	Positive	
8	Why do we go through the browser and not to the device	I would point it to the device then	Cause the user, the student uses a device, not directly to a browser it's just a piece of software	Pro	Positive	Then just connect to the browser directly
			because the first user uses the device, and on the device is the browser	Pro	Positive	
		we can connect the student to the browser, but then divide the device and operating system	Because we only depend on that there is a browser available, and it doesn't really matter what the operating system is. So then just leave device and operating system out	Pro	Positive	
9	Such like traffic light changing, but how to put it	so after running the simulation they still can change the parameters and then effect the simulation				you first need to set everything, and then you can run the simulation
		So I would say that you, you have the visual map then, then you have the behaviour. From behaviour you go to run simulation- so you would have a sort of loop				
10	how far do you want to go because if you don't watch out, if you don't- that you are not going to model- that you start modelling and, yeah, information flow, something like that, because- like in a FAM you run, you're looking at how modules interact right?	So maybe it's- right now we're looking from the user's perspective, of course already, like ok, you're gonna create some map, maybe we should think more in module terms. That for example. So, then we have only two components right, there is some kind of visualization, and there is a simulation. There				So maybe we can all first wrote those two top level elements and zoom in on both of them
11			the user can construct the roads into four way intersections	C	None	
12			then for the roads I have sensors and the roads can have different lengths which the user can specify	C	None	
13			then there are six default intersections available in the program	C	None	
14	I'm thinking about it might be difficult to involve an external server for the simulation, cause some parts have to happen on our server, and some have to happen on the external server.	I really- I think an external server or just a server on which the application runs. And all simulation also happen in that server	we now ofcourse assume that it goes to an external server and in fact we also should adopt this server. once the calculations happen	A	Positive	I think that maybe we should just set everything on the same server then
			Yeah I think that's easier	Pro	Positive	
		Do we have an external server for simulation calculation	may have some overhead of delegating the computing to another	Con	Negative	
			And there's also more latency involved right.	Con	Negative	

15	What does the simulation, have to do this, and there's some queuing	For example when you know, you have these traffic lights then for example right. Then cars are waiting in front of it, well you have multiple intersections, and every intersection has its own light right? So then you have of course, yeah well, for example, have two of them stay red on the same time, just saying some things, then you have a very large queue for example, right. Very large, well, a lot of cars of course, waiting for a red light, for example				lights are then, in fact related to some queuing
16	we can start, draw how they interact maybe with one another	user comes in, then interacts with animate traffic. Or don't? then of course- yeah from set parameters like this is choosing intersection. Everybody's choosing the section then				And also try to choose intersections and choose in fact what to do then, but- then they set the parameters after fixing the intersections, and then animate traffic
17	then the problem of that is that we have like, several modules for each thing then maybe	So then you might say like, ok just take one general module that takes into account all these properties based on the properties it can just calculate it.				Just leave that out then yeah
		So queuing, so maybe just simulation as the only module then , just leave it empty				
18	how to change the speed of the car					that's outside of our scope
19	I'm not sure exactly how these sensors work	so select sensor theme is a good one maybe, and then, yeah like, yeah because you were right about that of course, you have multiple factors which determine when a light goes green, of course.	But how about if we- if somebody touched the- somebody wants to cross the road and he push the sensor	Con	Negative	Maybe we shouldn't think that
			the user must have an option to add a sensor or not	C	Negative	
		it just keeps to its own timing scheme				

Team 9

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			they need to be able to change it during the game	C	None	
2			needs to be real-time	C	None	
3			We're not allowed to make crashes	C	None	
4			no intersection without traffic lights	C	None	
5			every intersection has a traffic light	C	None	
6			no roads without traffic lights	C	None	
7			all intersections will be four way	C	None	
8			your approach should accommodate at least six intersections	C	None	
9			all intersections are four way, so no T intersections. And also no one way roads	C	None	
10			And you need to be able to add sensors	C	None	
11			then of course you have an editor, with the roads. That you automatically- when you cross a road- because they're two way lanes	A	None	
12	we need to make a difference in the type of lights we use. That's also possible. Because if you say, to left and to the right, if you have a different light for each of those then you get a completely different flow and situation]	you have two kinds of traffic lights then right	your approach should also be able to accommodate left hand turns protected by left hand green arrow lights	C	Positive	I would change the traffic lights later in the process, because this- right now you can make the roads. Then you should already have a basic traffic light if it crosses a road. Afterwards, if you can get some kind of thing that you can click on, on the traffic light that opens a small box, and there you can change the traffic light. I think that's the second
		but I mean that you have a light with just a circle and that you can just drive, and that you can go left as well, without having a separate traffic light for it	what an unrealistic simulation	R	Negative	
			no I wouldn't- the more complexity you add	R	Negative	
13	I don't think we even have individual cars. Could be impossible	But is it possible then to model the- cause some roads are heavy traffic, some are not, and the main goal of the thing is that you want to simulate and see what the actions are	Yeah but we don't take that into account, because that's too difficult. Or I mean that's something we don't	R	Negative	So we don't have any driving time, everything we still need to do is how fast people can pass through a traffic light. We should incorporate that maybe somehow

14	you need to be able to predict which cars go left or right or straight ahead.	You just have a starting number on the side, and then you spread that number over the three roads coming in the intersection, and then you do this again, and again, and again. And each time you spread your number in the current road, or over the other three roads, and you take off the number of people of cars staying in the street. Then you add the number and that's the end-value of what's in your street. And you end value is just the sum of what came into the street from left, right and the bank. you need to know how many of those people driving, drove straight ahead. and how many took a turn. Unless you make a really elaborate box with You could also say, one third goes left, one third goes right, one third goes straight	that makes it all very difficult	R	Negative	they interact with the crossing information, that's where the numbers of the cars are then you don't necessarily need to it random, you can choose yourself and you can also say that you just let a certain amount of cars enter from each edge, just per minute or whatever
		So what I was saying, maybe we can put like, 100 vehicles on one side where they'll enter the map. And do some kind of random destination for every- car one needs to be here, and car two needs to be here, and just calculate the shortest path and it goes like that	Cause then it's also- it's a bit like normal traffic right. Because it rarely happens that anybody just keeps going and stays in the system forever	Pro	Positive	
			yeah that's really difficult, I think, because then we should first see were we draw roads and then we should assign destinations. Or something, no?	Con	Negative	
			Yeah. Wait is that extra or	Con	Negative	
15	we're now thinking about it, how are you going to spread the vehicles	If we do nothing random, and we let them just decide everything, then we have no random factor whatsoever and then they have full control	But if you do, the more they do the less they control the traffic. Of course	Con	Negative	I think we have to design the box. Because here's I think the most difficult part of the system.
		I think it's easier with the boxes and then just say, percentage left, percentage right, percentage straight, and then just keep counting up				
16			should be able to change the traffic density	C	None	
17			Should create a busy road or a seldom used one, or any variation in between.	C	None	

18	How do the- does the car spawner decide where the cars come, and then it spawns					This one is determined by one of these four, so one of these four is filled with this one. And you know which one by taking the location of the intersection
19	No accidents are allowed in the system, but somehow you have to answer to- manage the light for the traffic light	Either you have one which is just time controlled	No accidents are allowed in the system	C	Positive	I think that is how it works in real life, and that's easy to just make a screen and people can indicate how then does this traffic light work
			that's going to be difficult	R	Negative	
			with numbers it's easier.	Pro	Positive	
		Maybe you can make an OR like the light stays green for at least fifteen seconds, or 75% of the cars have to be passed	The traffic light will go bananas probably. it will crash I think, even, because with percentages it goes really fast. Because each time one car is less here, the other ones are going up, two times as fast.	Con	Negative	
20	The only thing is, with our model, in the beginning you're going to have an empty city.	maybe it should skip, like, two hours, when you press play. So the cars are already doing their thing	To make it a bit more realistic	Pro	Positive	Yeah I mean, that's not the biggest problem, I think. And the city will flow- will fill up in a realistic manner because you get the flows and you have the percentages so it's- you will get traffic on a lot of places where combined there are a lot of cars.
		you can fix that with either a random number in each street, or by just letting the simulation run for four minutes	NOT REALISTIC	CON	Negative	
			it should be possible to create a busy road	C	Positive	
21			It's probably Windows based	A	None	
22	But do we still have to make sure that it doesn't conflict with each other	check if the other one is green or not				you can choose which ones you want. And then you can say, this one doesn't have to be green that long, and this one does have to be green. Then you have a lot of stuff you can do.
		maybe just- the program prioritizes. That if all of them become green that you just take a random order				
		maybe you should be able to set some kind of repetition in it, so you get a minute and every 15 seconds or something, another light is green	Cause if you put them all in 15 seconds still then they can become green at the same moment	R	Negative	
			Some of the directions could be way heavier traffic then the others. And that would be a bit difficult or wrong to have only a short time frame for that one, with the same timeframe for-	R	Negative	
		you can just give them an order				

Team 10

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			it said here we have to have four way.	C	None	
2			student can enter the density for a lane.	C	None	
3	but you can have problem that two lanes are going into one lane now what about going from one lane to two lanes.	is not even. is no the best intersection.				never happens in reality
		not really. if he got two lanes then you always got two lanes but further in the road there go to one lane.	jam would start in the intersection	Pro	Positive	
			mostly intersection is symmetrical.	A	Positive	
			because there's no problem with traffic jam I think.	Pro	Positive	
4	mostly if you have this situation mostly the straight through car will only go to that lane and the other one will go there and then straight through lane and line on it I think I think you can from one lane to two it's not really.	we first start with intersection you place intersection on the map and then you can select road like a lane from a tool box and click on one side of these four sides of intersection and that you will add one more lane there and then program and the graphics will draw everything nicely.				it's allowed. so I suggest that type of construction.
		templates where you can select all the intersection. okay you always have four way intersection. and the you get all the option okay	but you can also have like nine lanes into one lane if you want.	R	Negative	
		so when he places the roads or just he can have this. the lane the road one will have two lanes and stuff like that. but then when you have all really construct that thing like something like this. and here on the two lane and here on the two lanes and here a lots of lanes. then you have a tool connect and then you say ok this one this connected to this. that means that here is then draw arrow and all the cars are in this lane are going only to this lane.	it should be easier to yeah for the individual single for traffic light.	Pro	Positive	
5	if you look at that with the global overview of this grey roads and green grass around and stuff like that. it is hard for you to see a little traffic light there.	if you want to put a traffic light on it you the computer will in will zoom in into the intersection and then you have big have graphical image of it and then you can see the lights.	the screen will go lets say black and you will see only purple traffic lights on each lane and green lane. so it's easier for you to focus only on this	Pro	Positive	I think that a nice visualization of it. we will have a different layout or different view of traffic lights. we will have separate view for traffic light system and then we have again this connection tool. traffic light connection tool. connecting tool. that works if you click on one light and the other light.
6			all intersections are four four way intersection	C	None	
7	so we can have roads that just end in the intersection.	so when you click run simulation . it will attract all this constraints and all the rules. and program will say to you. now you cannot run simulation because this road ends now where you need to connect it to something . I think that is like the elegant solution to and it can also be like red all the time or have the red border	every roads need to connect to something.	C	Positive	we've got two views and so attract all rules, when you start the simulation and do not start simulation if something is wrong

8			students will actually know what to build how to use this thing. what . how to connect the intersection. how to connect the traffic light and everything.	A	None	
9	so we something has to be done about that	what I suggest is maybe make tutorial for them. it can be like video tutorial and it also be like in games				yes go for it.
10	how to connecting intersection.	so now again we have this approach of connection tool.				connection tool.
11	now we have two intersections then. we have to know	I think that's most be built by roads so you track a road on it.	yeah I want to avoid this road drawing because it is really hard problem. we had one lecture on that and there is whole bunch of mathematics behind the road drawing	R	Negative	so you start one point and then you can say ok just drag a things on it.
12	should this still have this automatic joint calculation rule					yeah of course.
13	ok the problem is. Information like traffic density.	if users can change the lane maybe. like if he have full lane then he can not change from one lane to other lane. and if he has the (inaudible) lane then you can change from one lane to other lane.	yeah maybe just a little bit out of our assignment.	R	Negative	and I don't know if we really need to looking to that now. I think most a most basic for us more like ok when you need to design traffic light and not real simulation on it I think. so lets focus little bit more on the lights and little bit less on the traffic and how it responses to everything.
		it's simulation. but we can also we can stick on one lane and that's it.	no I don’t think that's that's important because everything is just based on some distributions it doesn't matter if well every rule will be incorporated in this system	Con	Negative	
			you know if you see a complete jam in one lane then you will switch lane and drive forward.	Pro	Positive	
14	and that it's written that every lane should have option to. students should be able to change the traffic density. that enter the map on the given road so enter the map so source.	we need to have some lets say circle that you place on the road and that is the source of traffic.	students should be able to change the traffic density	C	Negative	and nice yeah ok if you click on the source as well then you see nice scale ok this is real dense.
15	they don't know the direction.					they can specify the driver behavior.
16	how to visualize cars.	just dot .				I think that's dot are good enough
17	first the problem is that as I said when we're talking about traffic every car. every driver should have it's own logic like that should be program. that's like driving and considering the speed limit and stopping when it's red. and going forward when it's green. and following all this rules	have a separate class for every driver	because in the simulation you do not need to see. you only want to see how traffic .	Pro	Positive	we'll just say every car has it's own logic.
			yeah it's a nice visualization.	Pro	Positive	
			isn't that a little bit too specific on map programing.	Con	Negative	

18	students must be able to describe the behavior of the traffic light.		student must be able to create visual map	C	None	if you click on the traffic light then you get another pop up when you can set the variety of sequence and time schemes.
			students must be able to describe the behavior of the traffic light	C	None	
19	okay the problem is. how we program the traffic light basically.	so we need for that. we need a variety of sequence. timing schemes. and sensor.				oh basically we just have the class of all the rules in it. and there is some algorithm that will determine when to open some lights and how to determine if the rules are contradictory or something like that.
		you can also connect sensor from one lane to traffic light another lane.				
		have master traffic light and say the slave okay. and then we have two sensors. and this traffic light which is the master is connected to the	I don't think need slave and is more like okay this two are just one head traffic light.	Con	Negative	
			that add additional complexity	R	Negative	
		the thing is the sensor should now connected to traffic light at all it should be connected to some central unit.				
			better constraint the light wouldn't have. the light have constraint okay they can go both green at the same time if there is. if there is traffic on it. coz you have four way lane and if you go okay. if you go green and straight green, and this one is straight green that one. you can't coz you cause collision	C	None	
			I believe there are some algorithm that would solve this problem. and I don't think that this is actually the thing that we should discuss now.	A	None	
20	the system be install or web base. like a stand alone or inside cloud.	a stand alone	I think stand alone is easier.	Pro	Positive	okay so we go to stand alone.
			it's definitely cheaper.	Pro	Positive	
			cheaper for the company develop this software because simulations are really expensive for processor and memory power	Pro	Positive	
			it's heavier to maintain.	Con	Negative	
			yes. coz you have to update everything.	Con	Negative	
		inside cloud.	to maintain. oh that's not true. true	Con	Negative	
			and if you're not online then you can't use system.	Con	Negative	
			if you have like thousand people running simulation in your server you will need to have really really good server and that cost a lot	Con	Negative	
		I think you can do both.	but that consume al lot of money and time.	Con	Negative	
			I don't think we have enough information on it.	R	Negative	
		when I think about everything I think that is cheaper and easier to have local stand alone version.	you can easily push a new update every hour if you want	Pro	Positive	
			there can be also an option to pay for usage of this server for every simulation or for every hour of simulation.	Pro	Positive	
			I think that quite heavy for a web base.	Pro	Positive	
			there can be an option. but it can be also very expensive	Con	Negative	
			coz that object oriented and it's it runs good simulation.	A	Positive	

21			that just let say only right traffic is not English traffic	A	None	
22			yeah. this is a constraint. only right hand traffic.	C	None	
23			if you actually building simulation for real life scenario. than it definitely matters. coz you wanna know which lanes are the busier.	R	None	
24	how to model the traffic light functionality inside the user view of functional view point.	I'm proposing to settings will be like traffic light logic management. and there will have everything from rule management for that one traffic light is select. up for so this rules are like for how long it will be open	because that will cause a crash and you don't want that to happen.	C	Negative	yeah. functionality only list of option and that's . variety of sequence what ever it is. timing schemes and sensor.
			I'm thinking what about if this is we going in too deep. we are discussing a particular scenario.	R	Negative	
			we could run into problem because we have a loop here.	R	Negative	

Team 11

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			it should allowed road of variant length to be placed	C	None	
2			varying length and different arrangement, at least six	C	None	
3			that car should not be able to collide	C	None	
4	if we were to design a feed back to the professor how the student perform	to do this you also have to building some measurement measuring.	and more in general the professor can se like how the students perform if she need to provide more theory about.	A	Positive	so yeah therefore I think we should skip it. maybe in the future .
			an assumption is that the professor only gain from the system is that she wants the student to learn her theory	A	Positive	
			but then to give back the feed back of the performance of the students. so an assumption to would to be handy to prove the system later on or the professor to gain more benefit of the system.	A	Positive	
			and that can make the system more complex	Con	Negative	
			but on the other hand. if the student gets some kind of result. professor can get that result as well	Pro	Positive	
			implementing the feedback performance of students.	A	Positive	
			because of the assumptions that we made it wasn't in the requirement I think it will become problem.	R	Negative	
			we can program everything but we don't enough time to program everything. so I think that for it will be problem for our time.	R	Negative	
			with the professor. she want the theory to be convey properly.	A	Positive	
			student just want fun usable graphic.	A	Positive	
			so she doesn't care about safety. or usability. or maintainability. she want just that her students get the general idea of with theory . so wants her theory conveyed properly	A	Positive	
			fun usable. and they want to see some nice graphic maybe	A	Positive	

5	okay. how would you slice this.		road variant length and different arrangement of intersection	C	None	yeah. that's how it works
			students must be able to describe the behavior of the traffic light	C	None	
			variety of sequences and timing schemes	C	None	
			they should be able to see the result of the changes	C	None	
		like I said before like first start with creating a map. and tell the traffic density. and then view them as separate. traffic simulation. I think that the basic flow				
		so you group one function of it's creating a map. you group other function in a put in density. then a calculation will be done I think. and then it will be visualize.				
6	this is the viewpoint of the end user. how would you?.	I divided them up between interaction and system layer				yeah.
7			so we are assuming that calculates we already have the model and thing that can just do calculate.	A	None	
8			I think we don't have enough time for that. to go really in depth and see how to calculate it	R	None	
9	how is it connected with each other. how is it related. how would you call relation	one map has zero to many traffic light.				six to many. and on the other hand one traffic light is in one map. one to one map.
		six to many. and on the other hand one traffic light is in one map. one to one map.				
10	we laid those to one traffic light you mean	yeah we relate them to. so in the intersection at least four traffic light.				okay. so twelve twelve.
		so that mean you have three. three lights for every intersection for every thing so that's six. twelve.	if they all have twelve . is much easier to direct traffic coz this arrow and this arrow could go at the same time and if there just two is no way this and this could go at the same time	Pro	Positive	
			but there all four way. there all just crosses you know there all four way intersection	Con	Negative	
			but I think if you look the perspective of the professor. I think for the theory to cover the theory I think it should be handy to have those option as well. and I think for the architecting is not more difficult. is only the cardinality we give it.	Pro	Positive	
			I think this will help the traffic flow faster	Pro	Positive	
			I think this will help the traffic flow faster but it's more complicated to calculate. is just more calculation to be made.	Con	Negative	
			yeah. if you let them choose six or seven a lot more complicated to calculate. it kind also distorts just a flow of data if you have all the same lights. you can just the only variables are length, road, timing of the lights and the sequence which the order (inaudible) if you also add how many light there are per traffic light kind a defeat the purpose. so you don't actually see result of your scheme. but it also how many are the light are there in the traffic intersection.	R	Positive	

11			they said something a T section or not	C	None	
12			your approach should also be able accommodate left hand turn should protected by left hand green arrow lights	C	None	
13			the student must be able to describe the behavior	C	None	
14			it should also be able to accommodate left hand turn	C	None	
15	six yeah too many. in the traffic it is density I think it is distributed?	direction and a flow				I think that covers both. route
		contain the direction and direction maybe or route				
16	maybe we can also like if something is wrong	when it's like a problem it turns red. so people can see whether the problem lies and then				alright. when there's a lot of cars waiting like a pop up comes up. with little exclamation mark.
		exclamation mark.	so this is a constraint for placing on the code that every car you see should be a separate object. which will be appended to the main .once the density increases	C	Positive	
17			at least six intersections	C	None	
18	one of the problems that we're facing is how to do all the calculation and stuff	are we gonna do all in the main	we think that is easier far to do in the main	Pro	Positive	yeah well to give solution on those risk I think is just a small program I think it still
		that we create separate module to do all the calculation	risk is that we are creating code to large and complicated.	R	Negative	

Team 12

Event Number	Design Issue	Design Option	Argument	Type	Effect to DO	Decision
1			we need to make an assumption that is in the Netherlands, because traffic always every where different	Assumption	None	
2	I'm not sure how	country dependent.	also a constraint because your limit your self to the Dutch driving laws.	Constraint	Negative	being a primarily for a course at UCI, in the first version I think is efficient to make it specific for the united states .
			constraint is region country	Constraint	Positive	
		you can either chose to make it generic and make the rule set for region pop able so you can do region for Germany and for Holland and for many other regions.	so the trade off being that make it modulear will be more expensive	Risk	Negative	
		you can chose to just limit your self to the Netherlands and that's it.				
		it's the university of California Irvine. so I think we should deliver our self in to the united states then.				

3			user language is also the same like for the traffic rules it's will be American English.	Assumption	None	
4	how the application will be deployed	desktop application	old fashion and very problematic if some people have mac and some people have windows. then you have to go through all problem do java application.	Con	Negative	so I think we should make web application.
		web application	because in app you have same problem with different platforms	Pro	Positive	
			if we implement the wrong business rules it will be avoid a traffic response for how the traffic rules are we are and then the whole application is useless and this is I think the biggest risk.	Risk	Negative	
			the business rules being wrong.	Risk	Negative	
			wrong implement and then the system was useless	Risk	Negative	
5	what do you do we want like a lecturer can change this or the developers can change who should be the authority and the context that can change this.	Ideally you make the system somewhere that professor can make rules himself that I think it might be very				I think we should make those rules very modulear so you can easily swap without other rules and that there not entangle in the rest of system so if you make a mistake you can easily fix it. okay. so solution for this problem would be to make the traffic rules dynamic and editable by end user being the professor.
		so we should look into that if that possible if you can make the rules set dynamic and so professor can make it at his own rules and constraints to the traffic light simulator.				
6	then we also have the database right	you can also store a lot of thing in the browser. so like when the user draws map and he changes the traffic light parameters. can. we can store that in web. browser.				so we have an external database.

7	so the question is. do we want to make them figurable or not	yes	how often is she gonna change the rules because if it's once a year. all the extra work of making such does it say anything on that in the assignment?	R	Negative	so I think we should take into account that later on we would make an editor for such business rules and take into account in the way we structure our code and then.
			I would suggest yes. because the chance that you do that we implement something wrong is quite big in the first time and then we need the version with feedback from the lecture to make it to maintain it and if you do this way if this way you need so much less maintain that almost.	Pro	Positive	
		not	I know what you mean. I am not sure if I agree. the question is how much work is it to make it configurable up front and those it out weight that cost of updating the rules based on the feedback from the teacher afterwards	R	Negative	
		I think we can add such a layer on top of it. later on.				
8	are there more steps in hold of creating the map		intersection. creating roads of varying length.	C	None	I think they should only capture the high level of flow a user through the application I think it's in context viewpoint
			allowing roads of varying length	C	None	
			accommodate at least six interactions of intersection	C	None	
			you also need to design if whether each intersection has the sensor or not.	C	None	
			accommodate at least six intersections if not more	C	None	
			every intersection had traffic lights I believe	C	None	
			all intersections will be four away so there's no run a bus and there are no T intersection or one way road.	C	None	
			what does mean by readily that is already there by default	A	None	
			all intersections will be four way	C	None	
		you can maybe save or load the map or export it to as a pdf or something like that the student goods.				
		run the simulation				

9	how or you gonna know that something indeed is better than something.	that's why I said to import a specific map from like the lecturer and on that map you can high score because it's quite if you only have one road. that's a lot of complicated stuff to implement.				so I will just put another functionality with the question mark
		you can also implement something that analyze the amount of delay and then gives feedback to the user based on how well you perform in that moment. so that would be an extra.				
10	yeah I'm not sure we should put import and export module	into different modulees				so you draw a box around it.
		one modulee.				
		can you make a module of a map management or something like that and thought in that module there are is the import module and the export module. so you just throw a box around that so we know to same category. but how do we say it is just import and export and those are.				
11			with regard the functionality we have made the assumption that it would be too work for to make very broad rule editor so we have left out the rule.	A	None	

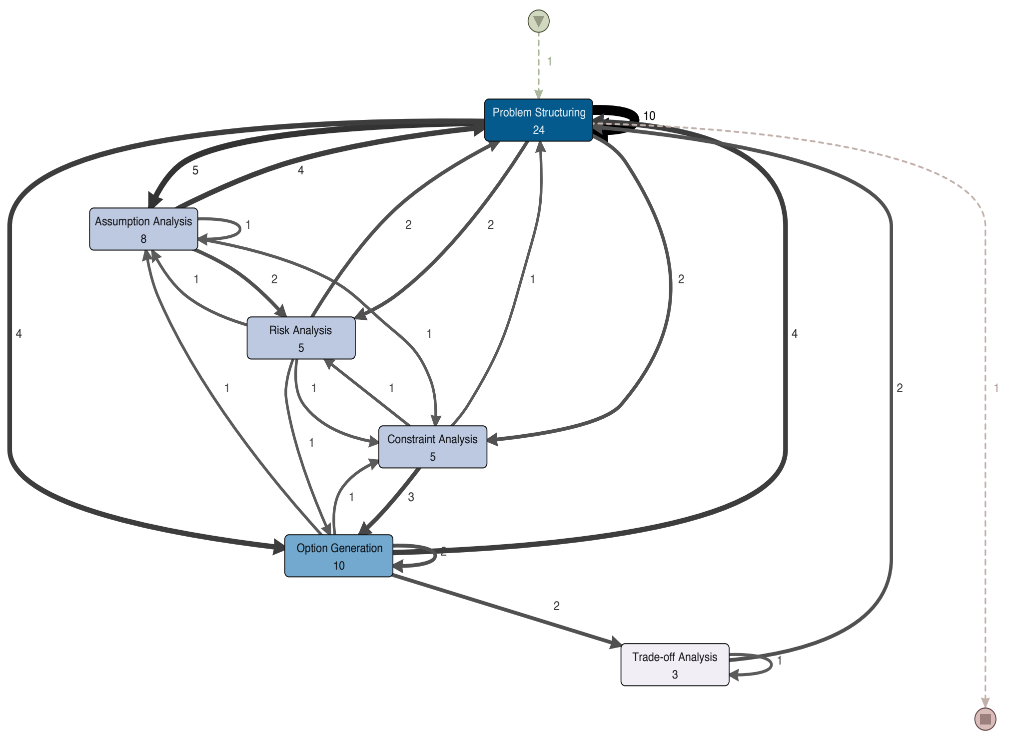
12	you have a like time make time interval something like.	yes so we updated every one hundred mille seconds or something.	yes like that and I don't know what reasonable in this thing but.	R	Negative	doesn't need to be one hundred mille seconds or something like that. not should be thirty frame per seconds and that would be fine.
		I don't think it needs much as much as frame per seconds a game like all they do or something.	yeah it can be even less if the performance is not very good because well it's just a simulator it doesn't have to be high.	Pro	Positive	
		to be simple and it's a web application and so don't make it too hight fidelity.				
13	I thought about in this thing last thing in about functionality maybe, there is help function or a introduction to how it work or introduction movie or do we want to include this? where were you included?	yeah it is in design time or in run time				yeah ok. just support we can put it.
14	yeah if he does something wrong how would does the system react to that.					you show an error and you can link to the support module which we say like.
15			combinations of individual signals that would be something crashes should not be allowed	C	None	
16	so we have a problem. and the problem is that some features are used in both feature.					perhaps we should not make high level distinction between design time or run time. but focus on modules and then later on we can always move specific classes.

17	so I have a problem what if someone changes the traffic light settings while a simulation is running. we want to keep track of the state I think or do we want to keep track of this state of the map or doesn't it matter and just. so to be clear someone starts simulation and changes some traffic light settings then the simulation will try at will change and because we have model it currently as reference from the simulation to the map. the traffic light setting will always change. you can't restore the original traffic light setting.	map version is copy of the map. so a map version		okay so put it in the map editor .data time element thing. should we put data element already but we should put also the functionality of map editor.
		action and lock or something like that.		

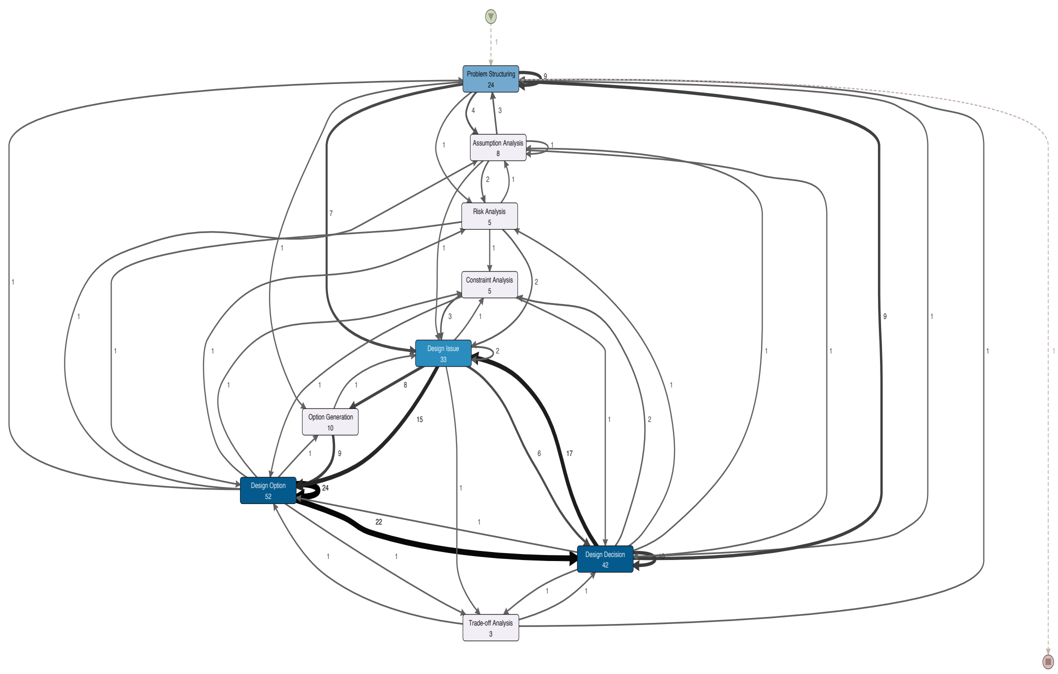
APPENDIX D

Process Model

Process Model of Team 1

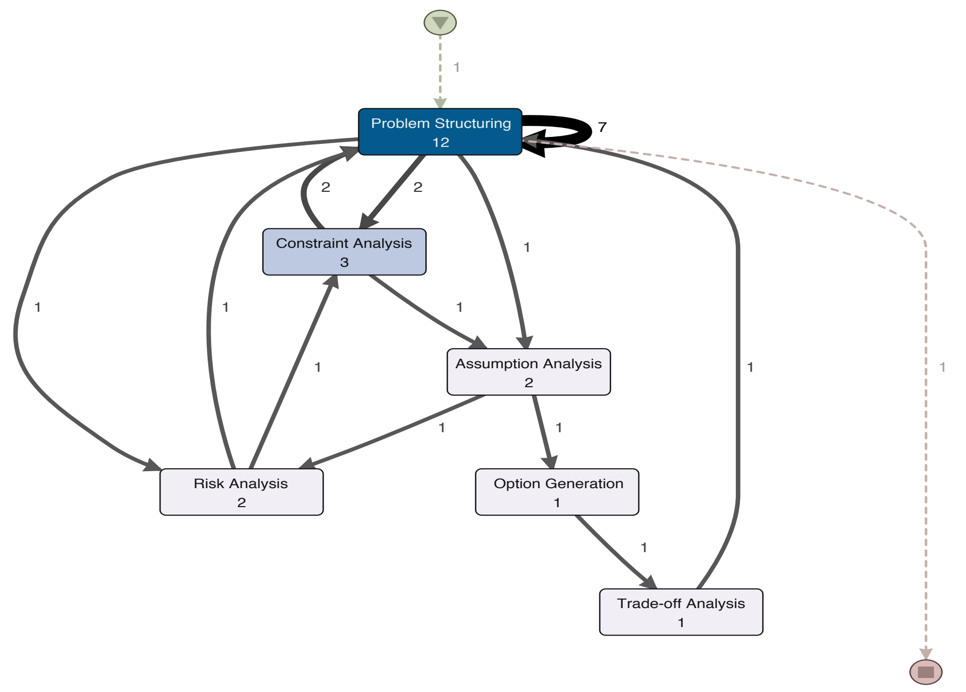


Process model 1. Team 1 reasoning technique

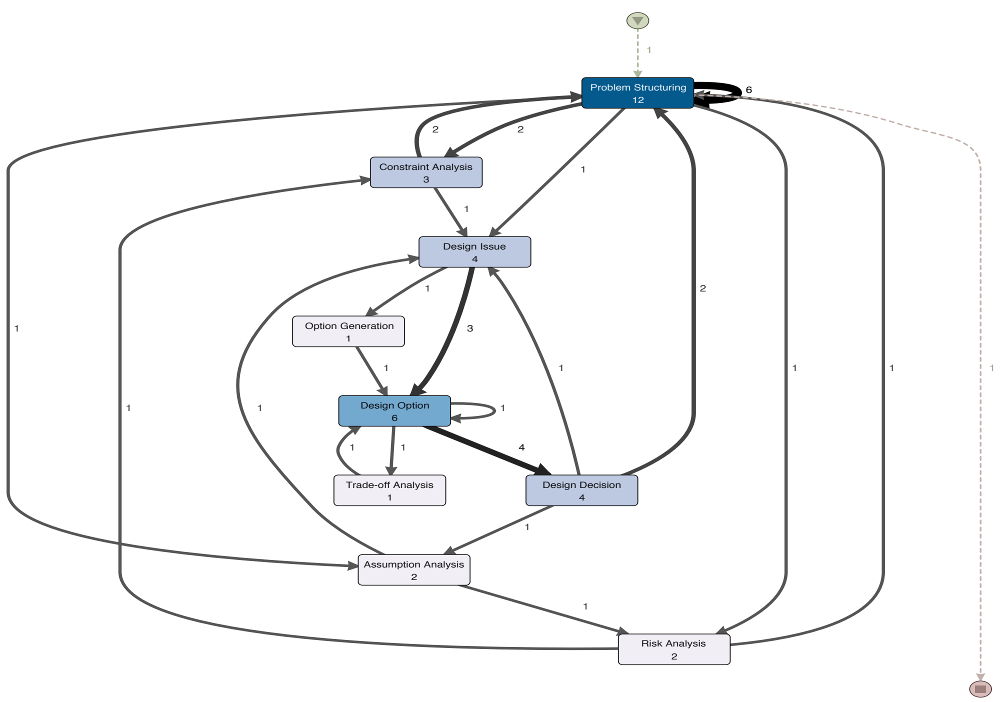


Process model 2. Team 1 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 2

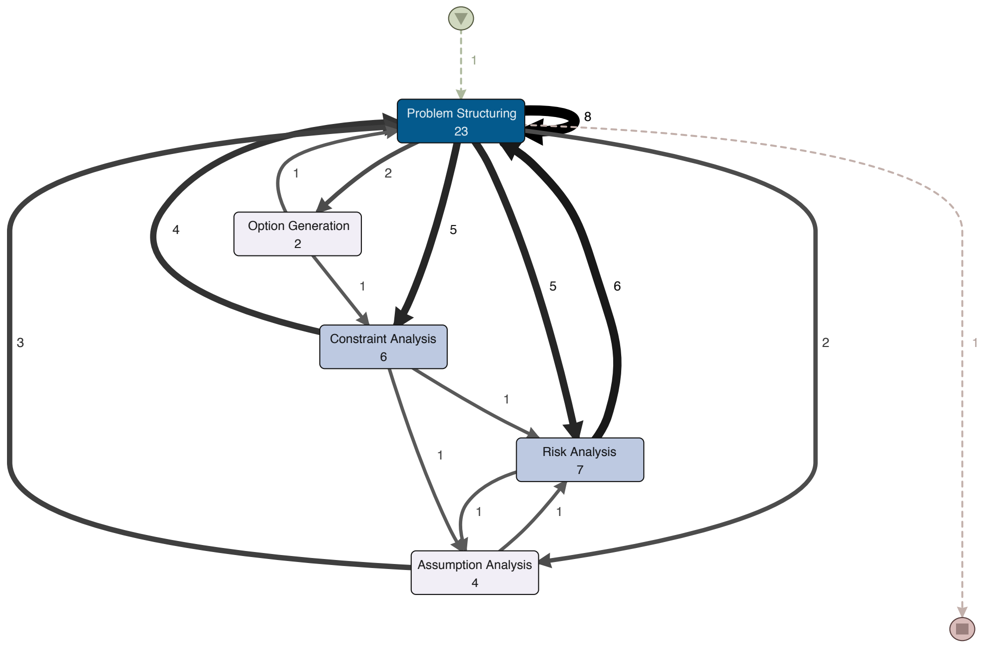


Process model 3. Team 2 reasoning technique

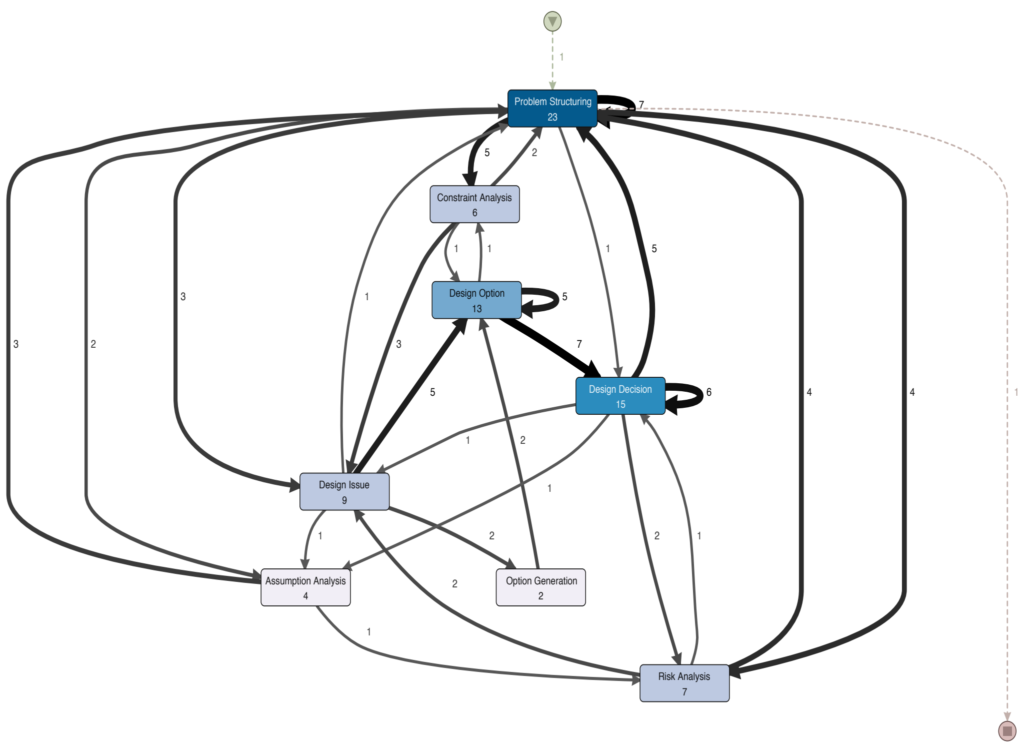


Process model 4. Team 2 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 3

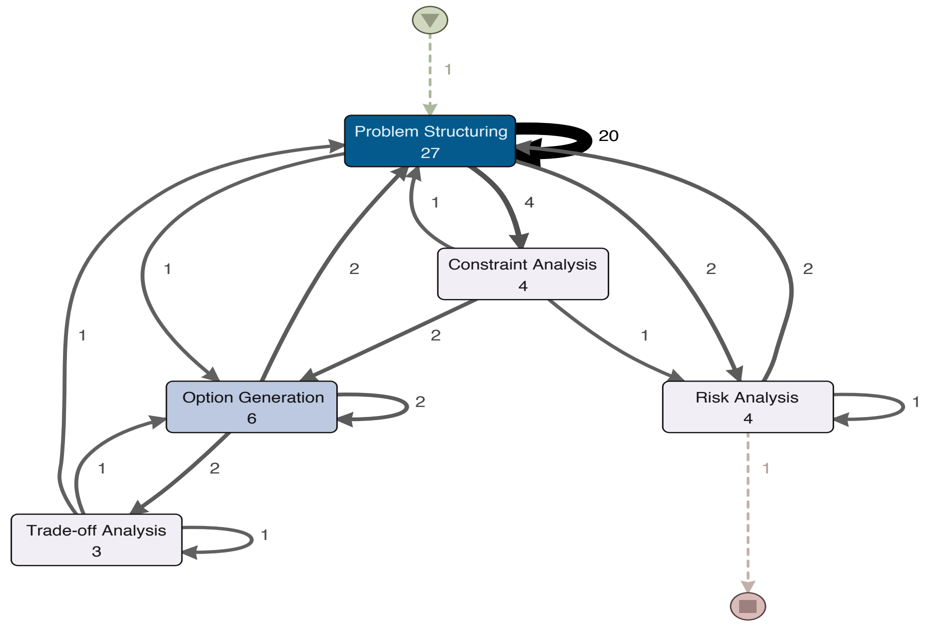


Process model 5. Team 3 reasoning technique

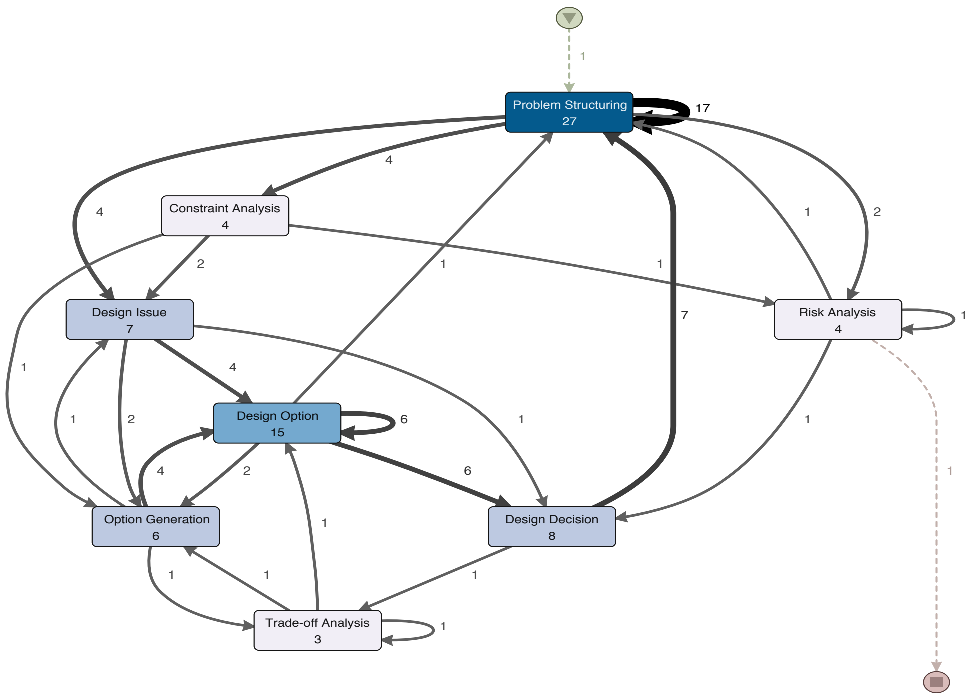


Process model 6. Team 3 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 4

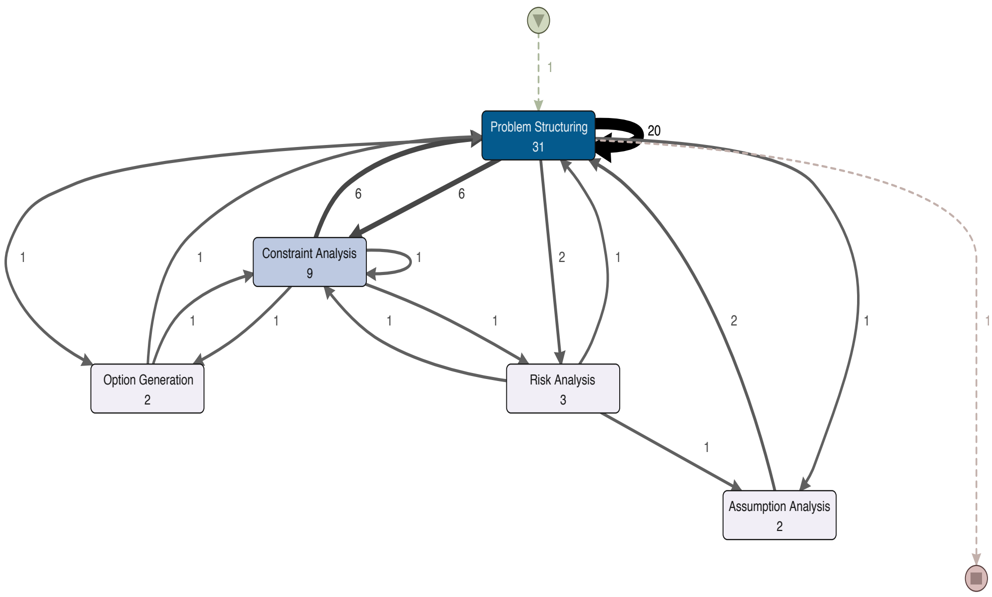


Process model 7. Team 4 reasoning technique

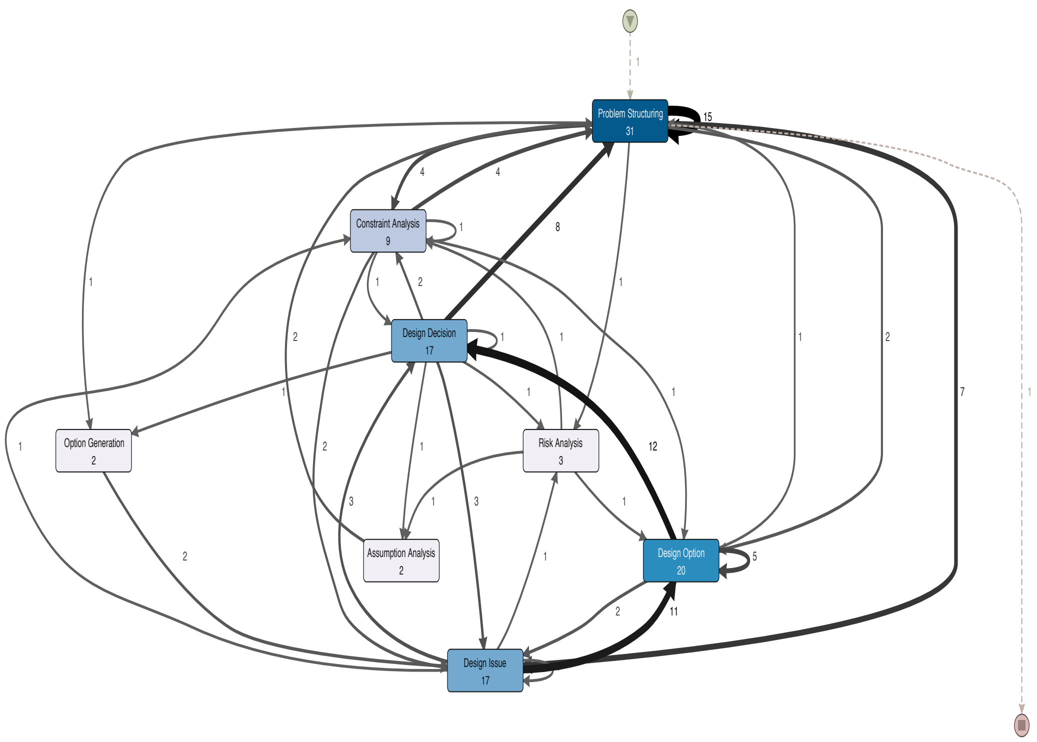


Process model 8. Team 4 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 5

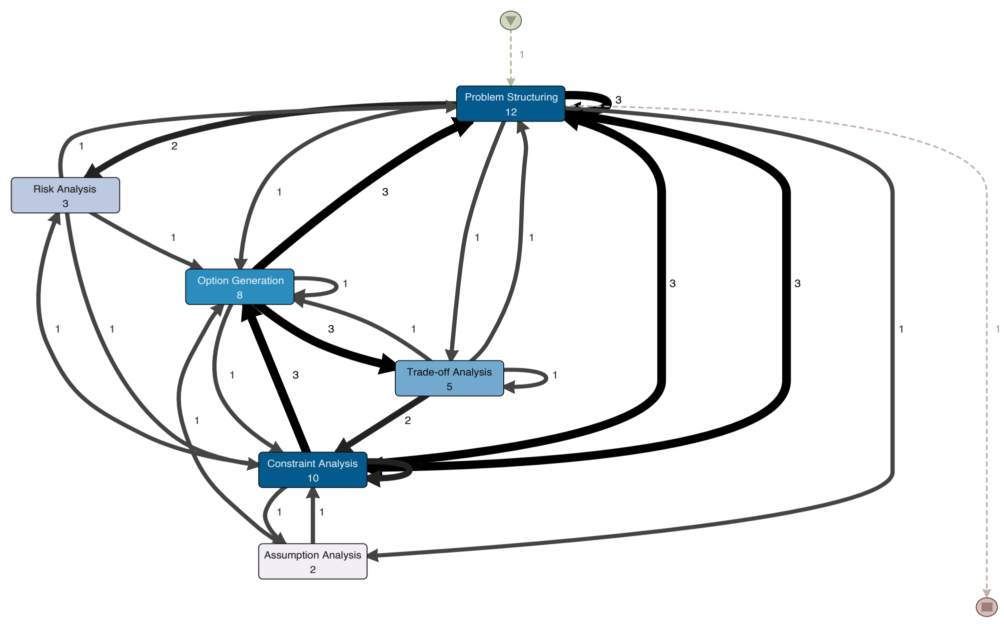


Process model 9. Team 5 reasoning technique

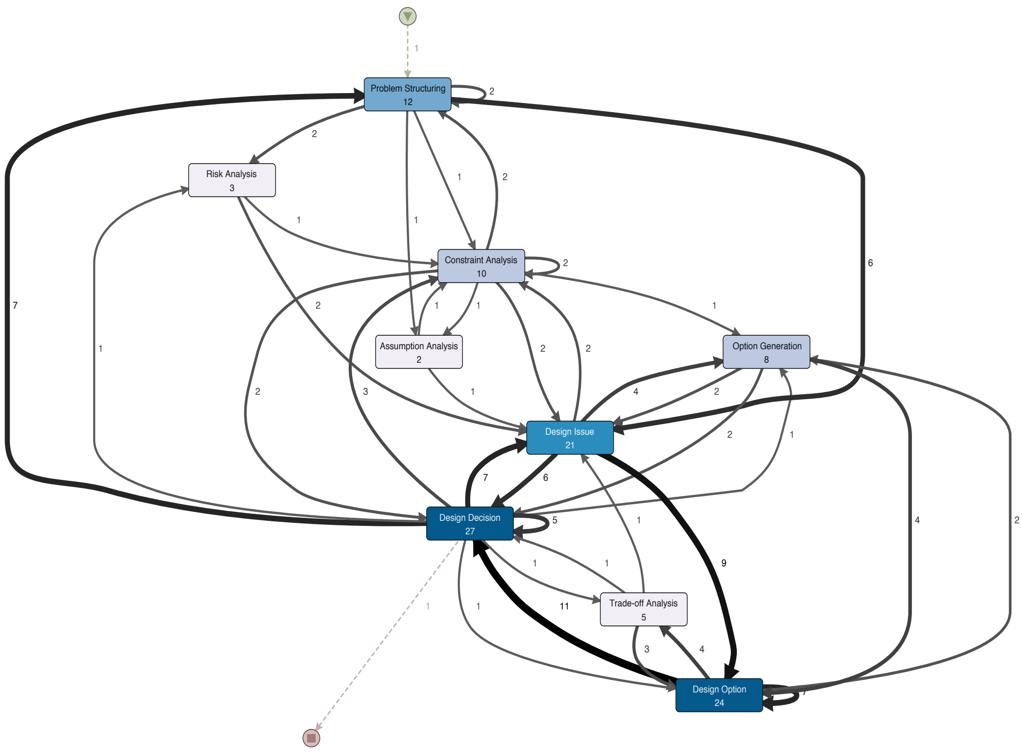


Process model 10. Team 5 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 6

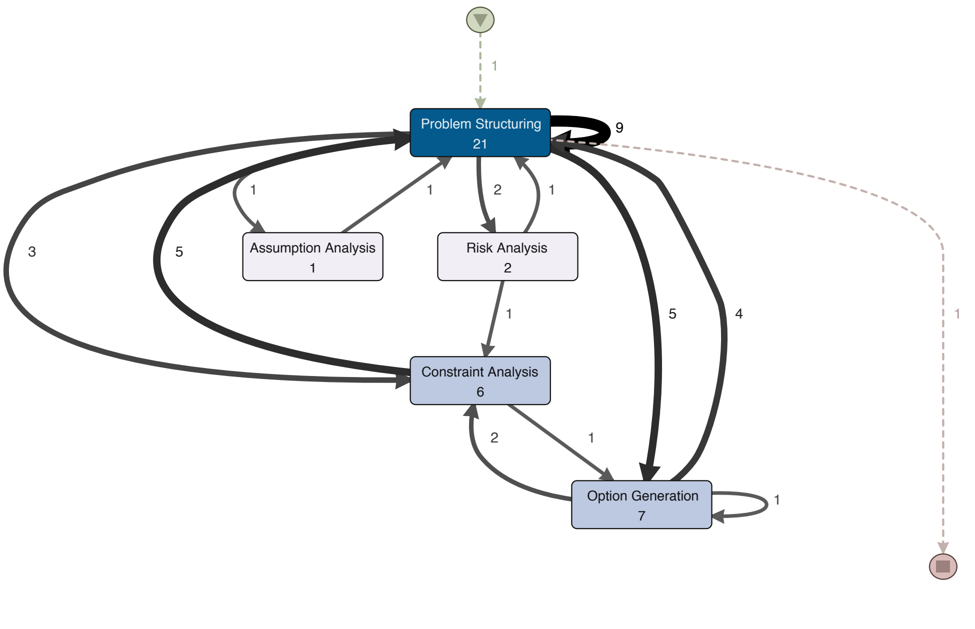


Process model 11. Team 6 reasoning technique

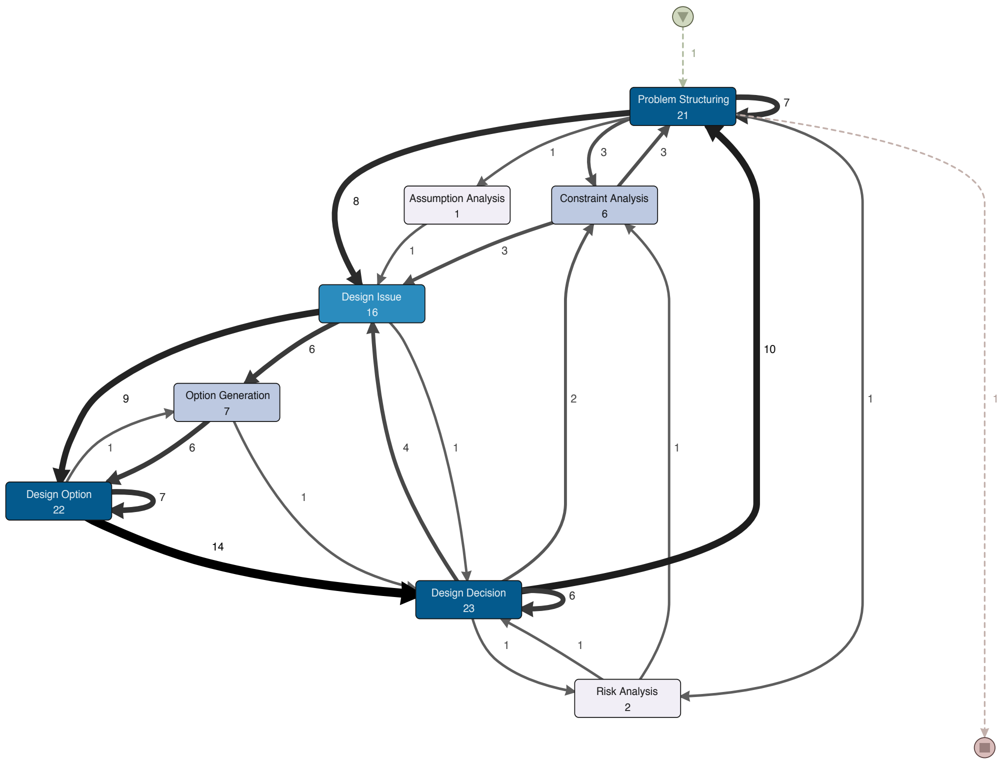


Process model 12. Team 6 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 7

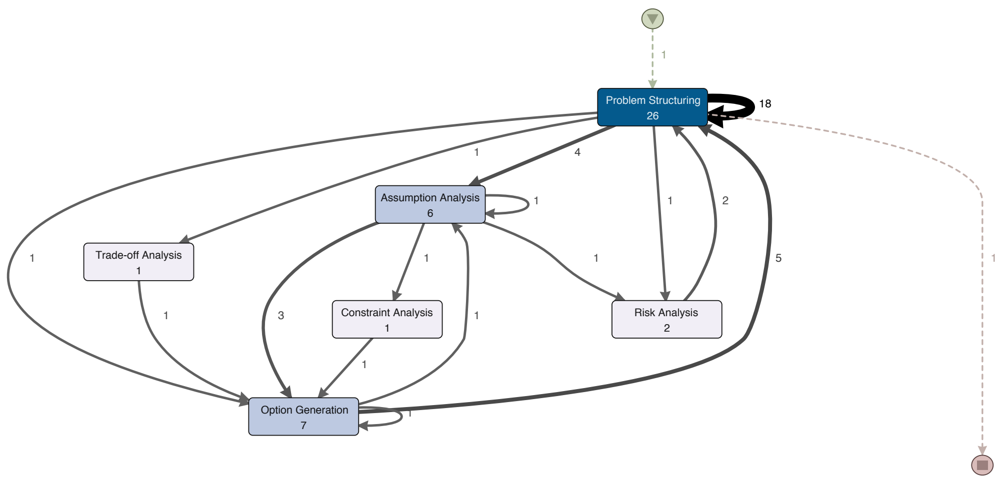


Process model 13. Team 7 reasoning technique

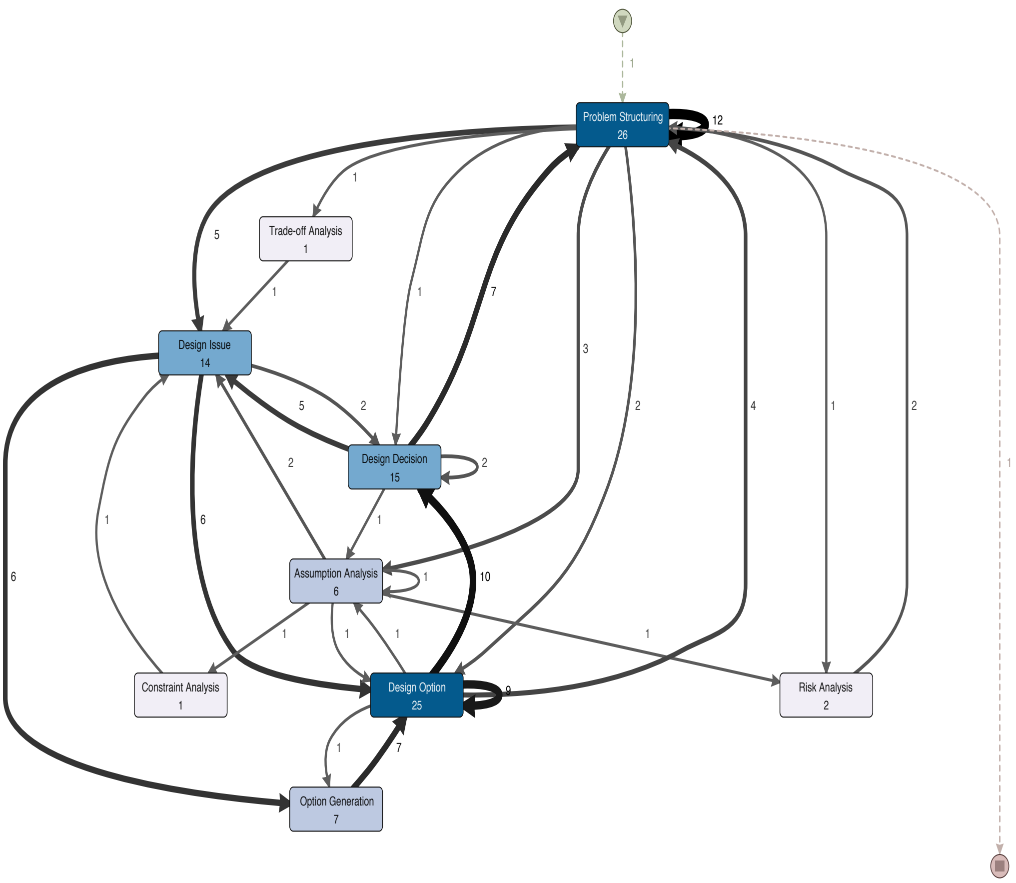


Process model 14. Team 7 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 8

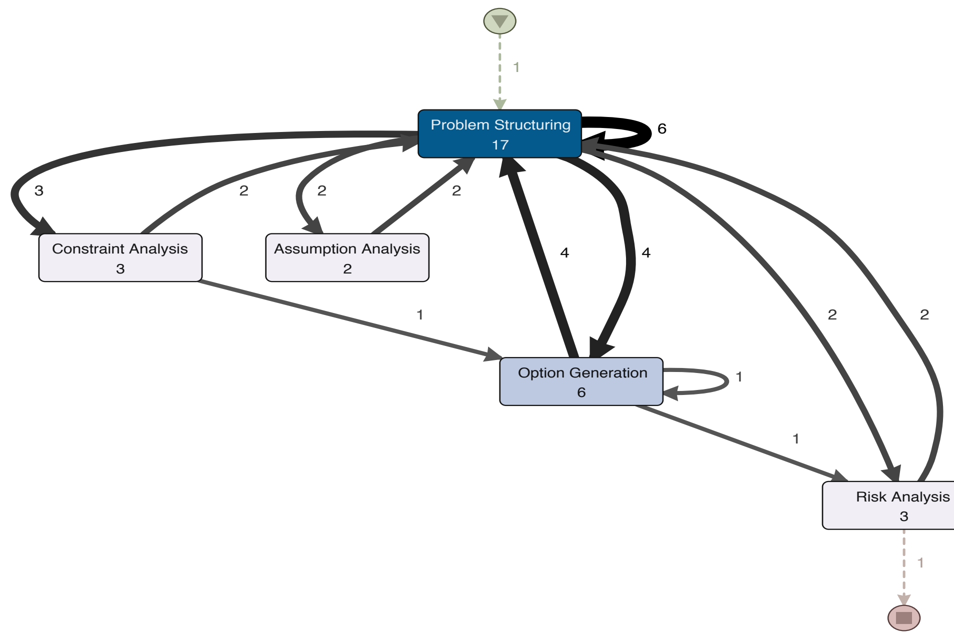


Process model 15. Team 8 reasoning technique

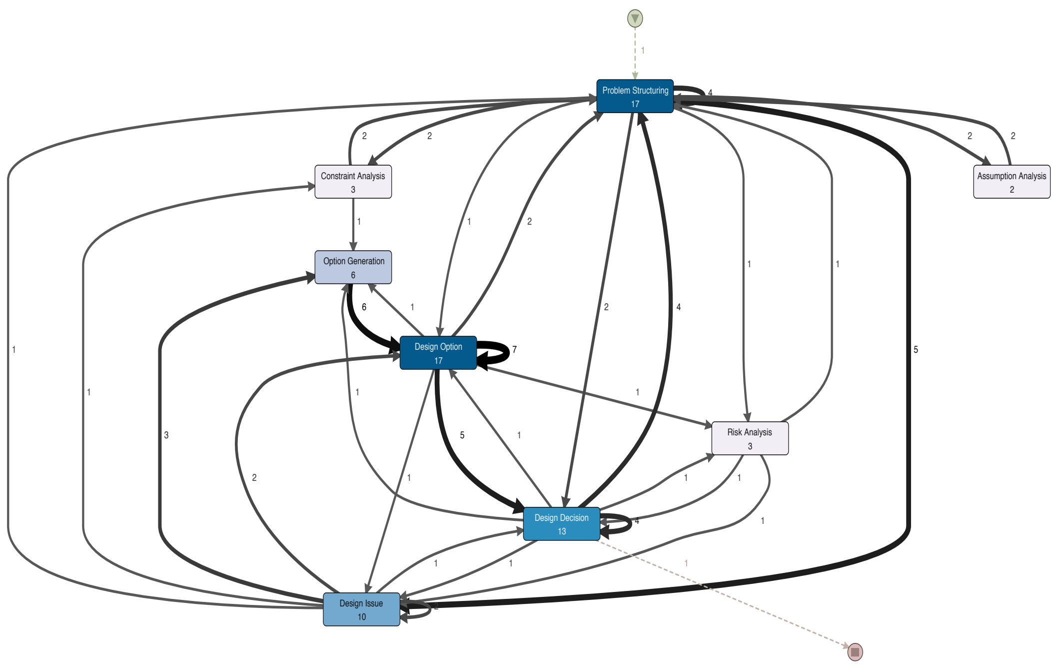


Process model 16. Team 8 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 9

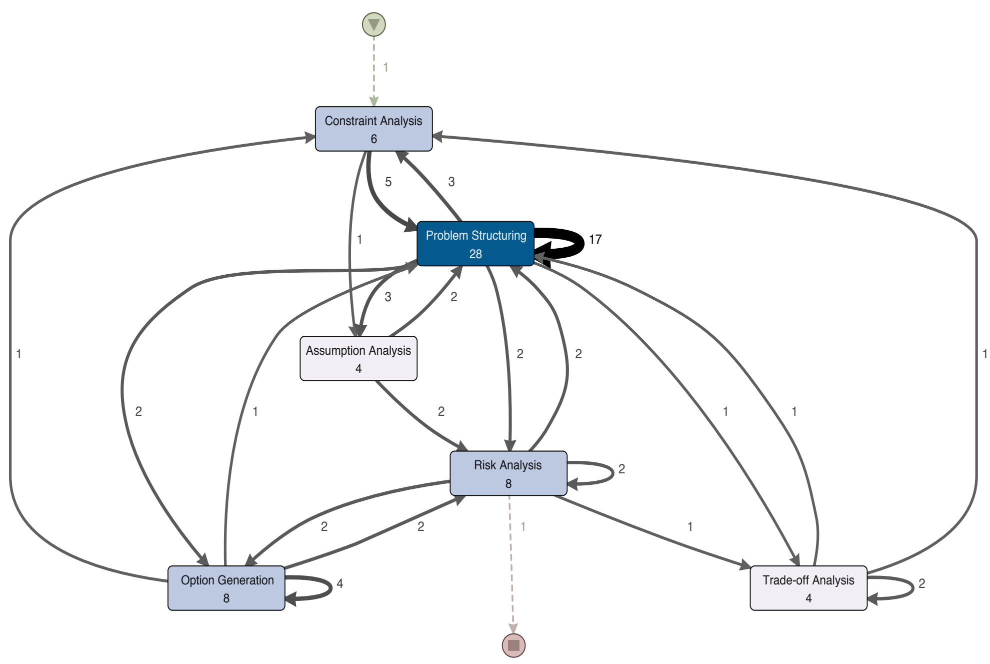


Process model 17. Team 9 reasoning technique

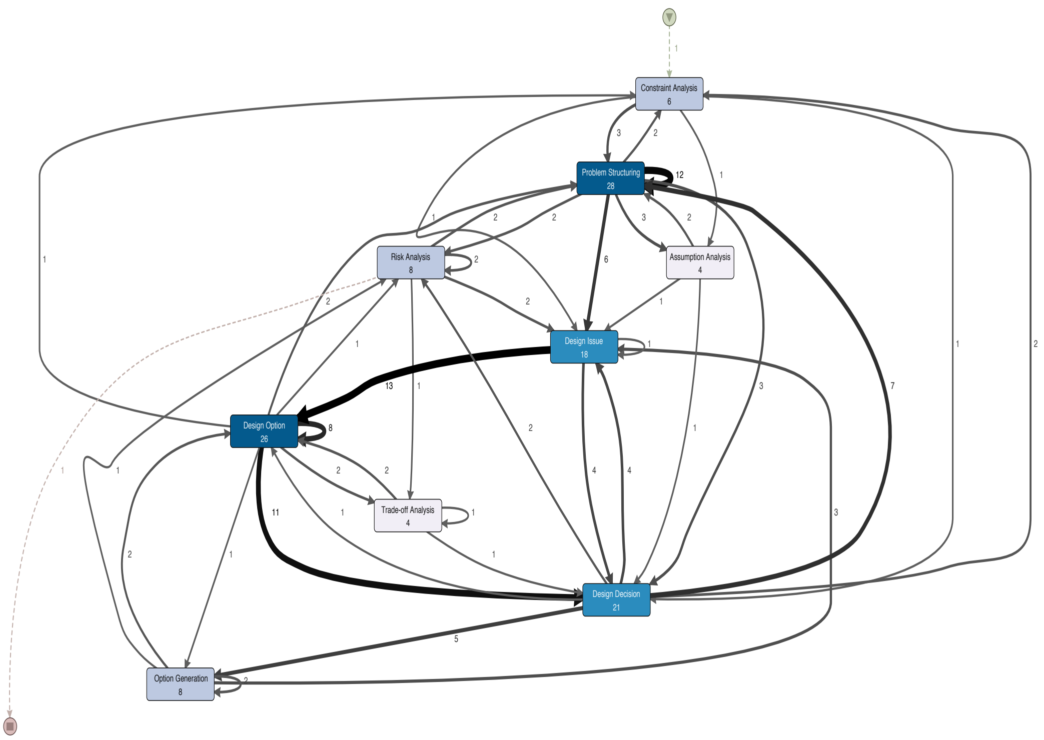


Process model 18. Team 9 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 10

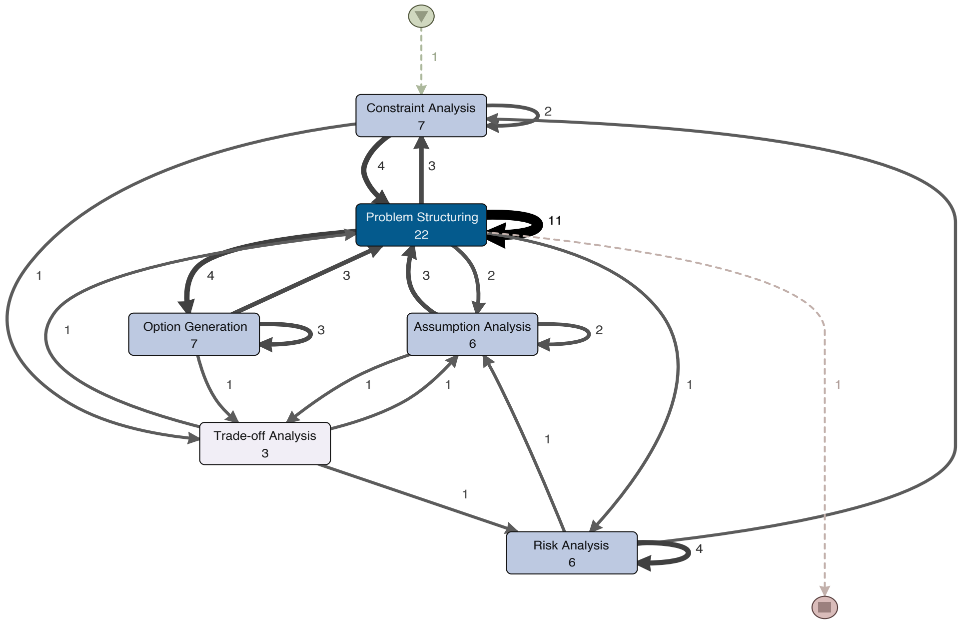


Process model 19. Team 10 reasoning technique

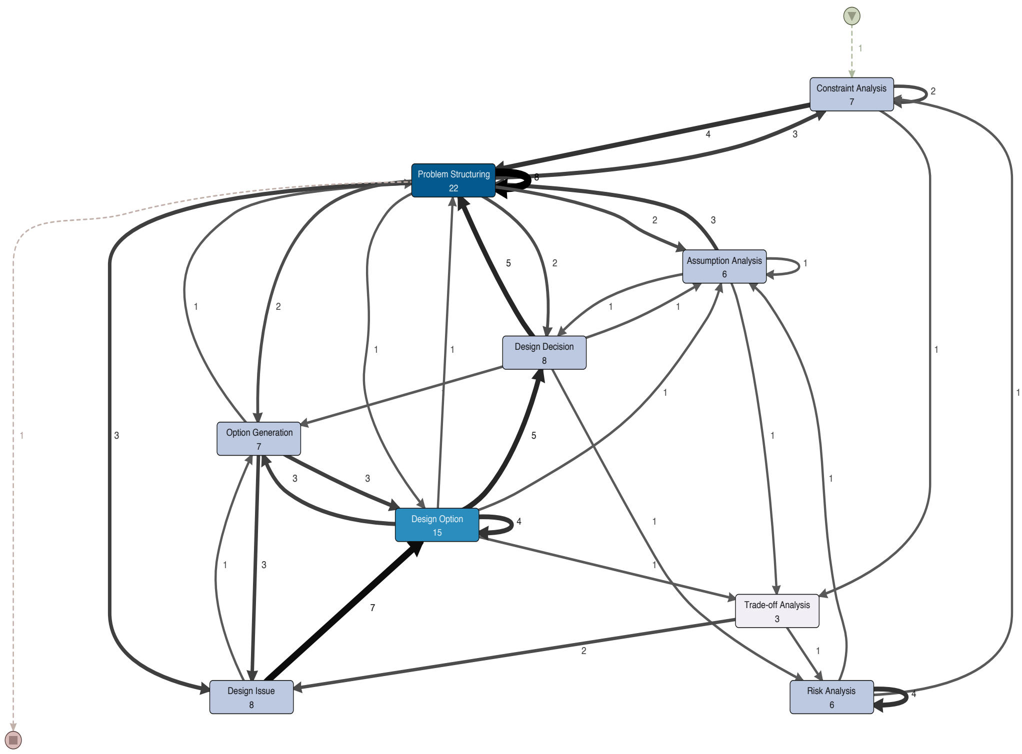


Process model 20. Team 10 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 11

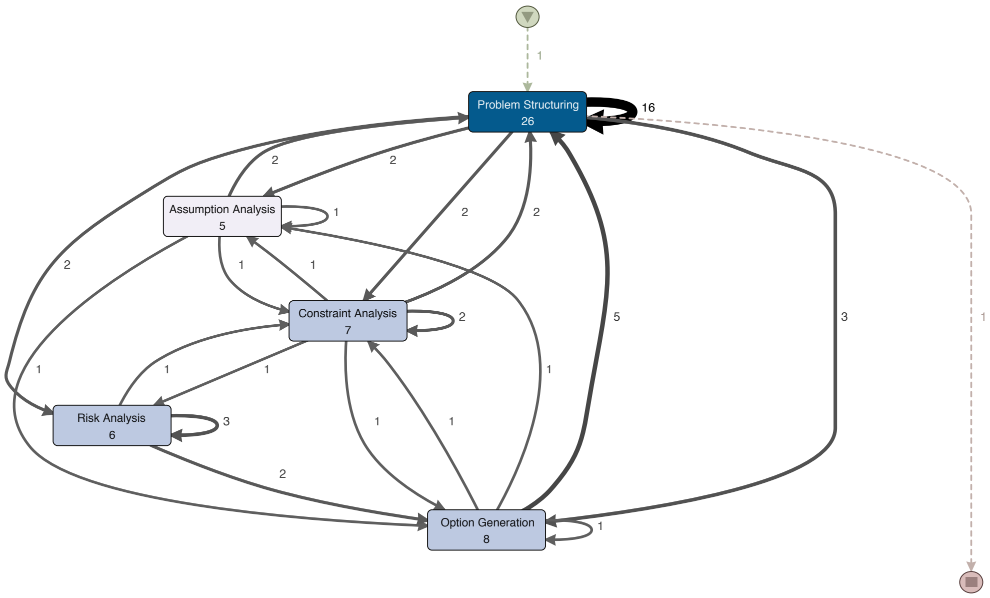


Process model 21. Team 11 reasoning technique

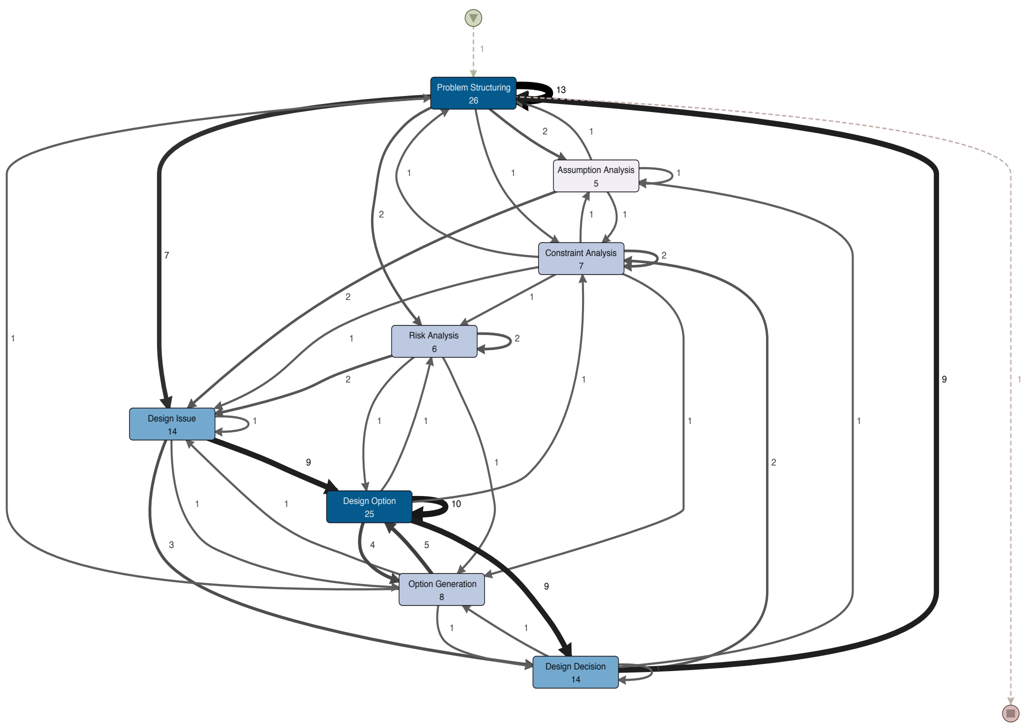


Process model 22. Team 11 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Team 12

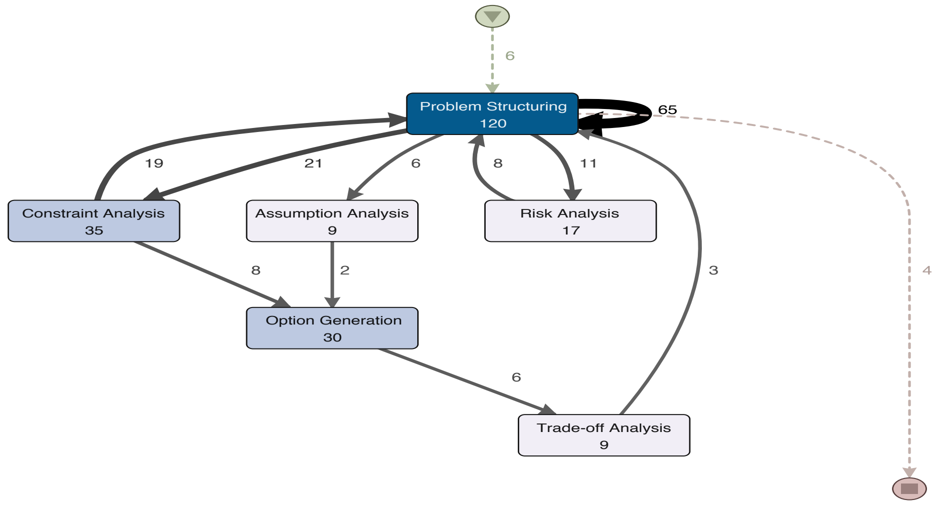


Process model 23. Team 12 reasoning technique

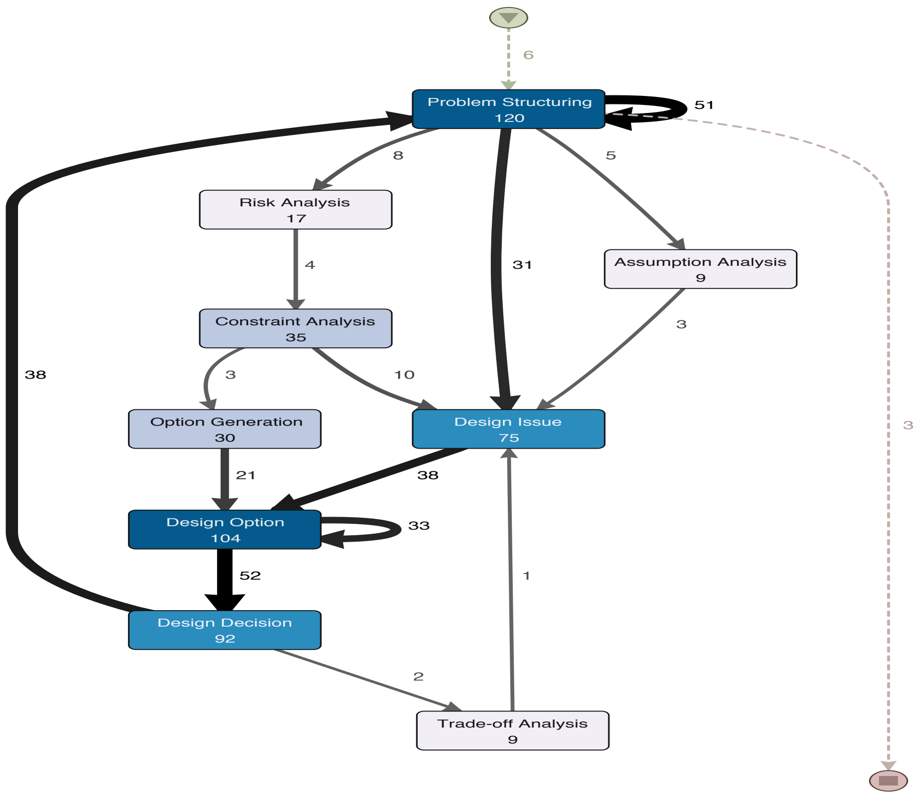


Process model 24. Team 12 reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Control Group

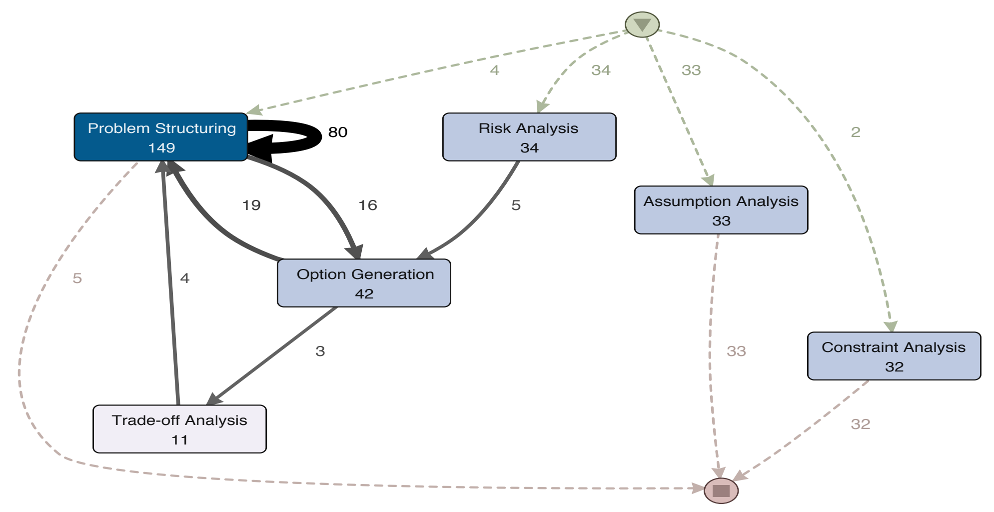


Process model 25. Control group reasoning technique

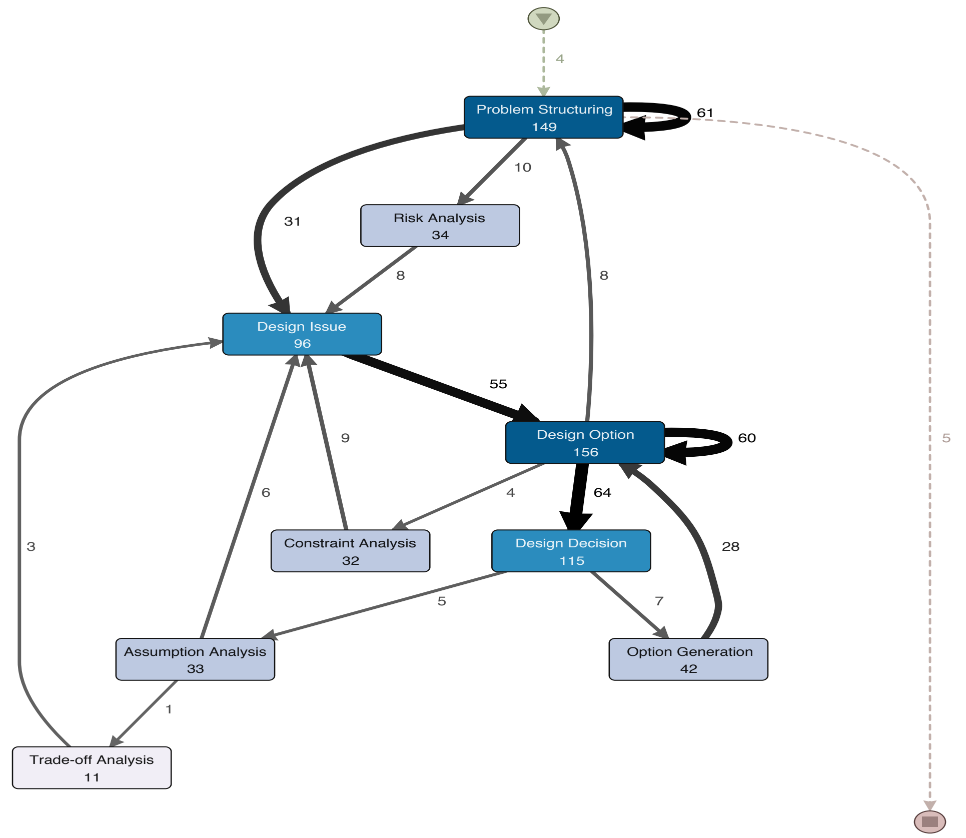


Process model 26. Control Group reasoning technique with Design Issue, Design Option, and Design Decision

Process Model of Test Group



Process model 27. Test group reasoning technique



Process model 28. Test group reasoning technique with Design Issue, Design Option, and Design Decision

APPENDIX E

The t-test Result

Assumption analysis	t-Test: Two-Sample Assuming Equal Variances		
Data Control group:		*Variable 1*	*Variable 2*
2, 0, 2, 2, 1, 2	Mean	1.5	5.5
M = 1.5 , STD = 0.84	Variance	0.7	2.3
	Observations	6	6
	Pooled Variance	1.5	
Data Test Group:	Hypothesized Mean Difference	0	
8, 4, 6, 4, 6, 5	df	10	
M = 5.5 , STD = 1.52	t Stat	-5.656854249	
	P(T<=t) one-tail	0.000105226	
	t Critical one-tail	1.812461102	
	P(T<=t) two-tail	0.000210452	
	t Critical two-tail	2.228138842	

Constraint analysis	t-Test: Two-Sample Assuming Equal Variances		
Data Control group:		*Variable 1*	*Variable 2*
3, 4, 9, 10, 6, 3	Mean	5.833333333	5.33333333
M = 5.83 , STD = 3.06	Variance	9.366666667	5.06666667
	Observations	6	6
	Pooled Variance	7.216666667	
Data Test Group:	Hypothesized Mean Difference	0	
5, 6, 1, 6, 7, 7	df	10	
M = 5.33 , STD = 2.25	t Stat	0.322375708	
	P(T<=t) one-tail	0.376903364	
	t Critical one-tail	1.812461102	
	P(T<=t) two-tail	0.753806728	
	t Critical two-tail	2.228138842	

Option Generation	t-Test: Two-Sample Assuming Equal Variances		
Data Control group:		*Variable 1*	*Variable 2*
1, 6, 2, 8, 7, 6	Mean	5	7
M = 5 , STD = 2.83	Variance	8	7.2
	Observations	6	6
	Pooled Variance	7.6	
Data Test Group:	Hypothesized Mean Difference	0	
10, 2, 7, 8, 7, 8	df	10	
M = 7 , STD = 2.68	t Stat	-1.256561725	
	P(T<=t) one-tail	0.118735824	
	t Critical one-tail	1.812461102	
	P(T<=t) two-tail	0.237471649	
	t Critical two-tail	2.228138842	

Problem structuring

Data Control group:

12, 27, 21, 12, 21, 17

M = 20 , STD = 7.85

Data Test Group:

24, 23, 26, 28, 22, 26

M = 24.83 , STD = 2.23

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	20	24.8333333
Variance	61.6	4.96666667
Observations	6	6
Pooled Variance	33.28333333	
Hypothesized Mean Difference	0	
df	10	
t Stat	-1.451088725	
P(T<=t) one-tail	0.088695568	
t Critical one-tail	1.812461102	
P(T<=t) two-tail	0.177391137	
t Critical two-tail	2.228138842	

Risk analysis

Data Control group:

2, 4, 3, 3, 2, 3

M = 2.83 , STD = 0.75

Data Test Group:

5, 7, 2, 8, 6, 6

M = 5.67 , STD = 2.07

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	2.833333333	5.66666667
Variance	0.566666667	4.26666667
Observations	6	6
Pooled Variance	2.416666667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-3.156820749	
P(T<=t) one-tail	0.005106923	
t Critical one-tail	1.812461102	
P(T<=t) two-tail	0.010213845	
t Critical two-tail	2.228138842	

Trade-off analysis

Data Control group:

1, 3, 0, 5, 0, 0

M = 1.5 , STD = 2.07

Data Test Group:

3, 0, 1, 4, 3, 0

M = 1.83 , STD = 1.72

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	1.5	1.83333333
Variance	4.3	2.96666667
Observations	6	6
Pooled Variance	3.633333333	
Hypothesized Mean Difference	0	
df	10	
t Stat	-0.302891266	
P(T<=t) one-tail	0.384089891	
t Critical one-tail	1.812461102	
P(T<=t) two-tail	0.768179782	
t Critical two-tail	2.228138842	

Assumption Data Control group: 2, 0, 2, 6, 2, 2 M = 2.33 , STD = 1.97 Data Test Group: 8, 5, 7, 5, 9, 4 M = 6.33 , STD = 1.97	t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	2.3333333	6.3333333
Variance	3.8666667	3.8666667
Observations	6	6
Pooled Variance	3.8666667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-3.523321	
P(T<=t) one-tail	0.0027539	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.0055078	
t Critical two-tail	2.2281388	

Con Data Control group: 2, 3, 0, 11, 2, 4 M = 3.67 , STD = 3.83 Data Test Group: 11, 1, 5, 10, 3, 1 M = 5.17 , STD = 4.40	t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	3.6666667	5.1666667
Variance	14.666667	19.366667
Observations	6	6
Pooled Variance	17.016667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-0.629817	
P(T<=t) one-tail	0.2714678	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.5429357	
t Critical two-tail	2.2281388	

Constraint Data Control group: 3, 9, 14, 13, 20, 15 M = 12.33 , STD = 5.79 Data Test Group: 6, 13, 4, 10, 13, 11 M = 9.5 , STD = 3.73	t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	12.333333	9.5
Variance	33.466667	13.9
Observations	6	6
Pooled Variance	23.683333	
Hypothesized Mean Difference	0	
df	10	
t Stat	1.0084094	
P(T<=t) one-tail	0.1685175	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.337035	
t Critical two-tail	2.2281388	

Design Decision

Data Control group:

4, 8, 16, 21, 16, 8

M = 12.17 , STD = 6.46

Data Test Group:

31, 9, 14, 17, 8, 13

M = 15.33 , STD = 8.36

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	12.166667	15.333333
Variance	41.766667	69.866667
Observations	6	6
Pooled Variance	55.816667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-0.734144	
P(T<=t) one-tail	0.2398604	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.4797209	
t Critical two-tail	2.2281388	

Design Issue

Data Control group:

4, 8, 16, 21, 16, 8

M = 12.17 , STD = 6.46

Data Test Group:

31, 9, 14, 17, 8, 13

M = 15.33 , STD = 8.36

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	12.166667	15.333333
Variance	41.766667	69.866667
Observations	6	6
Pooled Variance	55.816667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-0.734144	
P(T<=t) one-tail	0.2398604	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.4797209	
t Critical two-tail	2.2281388	

Design Option

Data Control group:

7, 10, 19, 22, 22, 17

M = 16.17 , STD = 6.31

Data Test Group:

40, 13, 20, 24, 14, 25

M = 22.67 , STD = 9.83

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	16.166667	22.666667
Variance	39.766667	96.666667
Observations	6	6
Pooled Variance	68.216667	
Hypothesized Mean Difference	0	
df	10	
t Stat	-1.363103	
P(T<=t) one-tail	0.101376	
t Critical one-tail	1.8124611	
P(T<=t) two-tail	0.2027519	
t Critical two-tail	2.2281388	

Pro	t-Test: Two-Sample Assuming Equal Variances		
Data Control group:		*Variable 1*	*Variable 2*
7, 7, 9, 16, 2, 3	Mean	7.3333333	10.166667
M = 7.33 , STD = 5.01	Variance	25.066667	47.366667
	Observations	6	6
	Pooled Variance	36.216667	
	Hypothesized Mean Difference	0	
	df	10	
Data Test Group:	t Stat	-0.815463	
22, 7, 11, 13, 5, 3	P(T<=t) one-tail	0.2169002	
	t Critical one-tail	1.8124611	
M = 10.17 , STD = 6.88	P(T<=t) two-tail	0.4338005	
	t Critical two-tail	2.2281388	

Risk	t-Test: Two-Sample Assuming Equal Variances		
Data Control group:		*Variable 1*	*Variable 2*
2, 3, 4, 4, 2, 7	Mean	3.6666667	6.8333333
M = 3.67 , STD = 1.86	Variance	3.4666667	1.3666667
	Observations	6	6
	Pooled Variance	2.4166667	
	Hypothesized Mean Difference	0	
	df	10	
Data Test Group:	t Stat	-3.528211	
6, 8, 7, 8, 5, 7	P(T<=t) one-tail	0.0027315	
	t Critical one-tail	1.8124611	
M = 6.83 , STD = 1.17	P(T<=t) two-tail	0.0054631	
	t Critical two-tail	2.2281388	