CHALMERS
UNIVERSITY OF TECHNOLOGY

Leveraging Social Networks Constructed from System Engineering Repositories

Master's thesis in Software Engineering

Mazen Mohamad

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2016

Master's thesis 2016:NN

Leveraging Social Networks Constructed from System Engineering Repositories

Mazen Mohamad

Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
Gothenburg, Sweden 2016

Leveraging Social Networks Constructed from System Engineering Repositories
Mazen Mohamad

Supervisors: Grischa Liebel and Eric Knauss, Department of Computer Science and Engineering
Examiner: Miroslaw Staron, Department of Computer Science and Engineering

Master's Thesis 2016:NN
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2016

Leveraging Social Networks Constructed from System Engineering Repositories
MAZEN MOHAMAD
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Communication and coordination are key factors to achieve successful requirements and software engineering. However, it is very difficult to establish the right communication and coordination channels, especially in large organizations. We investigate the automatic construction of social network models from existing requirements and other systems engineering models, in order to facilitate communication. We conducted a design science research study, resulting in the approach Low-Cost Communication and Coordination (LoCo CoCo), which automatically creates and visualizes social networks based on selected system engineering components. LoCo CoCo creates social networks of real-life, productive systems engineering models at an automotive OEM, which we evaluated based on 12 surveys and 10 interviews with professionals who assessed whether LoCo CoCo is beneficial in their everyday work. Our results indicate that LoCo CoCo as an approach is feasible and useful to overcome existing communication challenges. Remaining challenges are the quality of existing social data and a wider adoption into daily systems engineering work.

Keywords: Systems Engineering; Communication; Coordination; Requirements Clarification; Empirical Software Engineering.

Acknowledgements

I would like to thank my academic supervisors Grischa Liebel and Eric Knauss for all their support and guidance throughout the project. Additionally I thank every participant from Volvo GTT and Systemite, and the Swedish Institute for granting me a scholarship to study my master program. Last but not least, I want to thank my parents, my wife Rim, and my friends for all their love and support.

Mazen Mohamad, Gothenburg, April 2016

Contents

1

Introduction

In the past few decades the software market has grown and has become a crucial part of many industries which did not use any software before. An example is the automotive industry, where software has become a very important part of the overall product since its introduction [1]. The products of the automotive industry are not only vehicles of transportation, but also vehicles of information, communication, and services [2]. A number of conducted studies [1, 3, 4] show that an increasing value creation in the automotive industry is realized by software, and that the majority of innovations in that industry will also be realized by software in the future.

Despite offering a great opportunity, software also causes many challenges caused by the nature of automotive systems which are highly-complicated, safety-critical, and real-time [2]. The increasing size and complexity of the automotive software system imposes an increase in the size of the team working on these systems. Automotive software systems engineers deliver cross-domains features, e.g. body, chassis, multimedia, driver assistance, and human machine interfaces [5].

Requirements Engineering (RE) is an important part of the overall software development and poor requirements engineering has been described to significantly affect project success [6]. RE requires a lot of collaboration among different stakeholders in a software project, making intensive communication and organization essential in the RE process. A number of studies [4, 7–9] reveal that RE is a major challenge in the automotive industry due to the increasing complexity of software in vehicles

Communication is argued to be the key to successful requirements engineering [10]. A number of studies have been aiming to improve communication during RE. Examples are the study conducted by Paech et al [11] where the authors created an information model for RE to achieve understanding of activities and the study conducted by Soltani et al. [12] where the researchers suggested close collaboration, a focus on communication, and an iterative approach to address RE challenges found in the AUTomotive Open System ARchitecture (AUTOSAR) ecosystem.

Social network analysis (SNA) is defined as a strategy for investigating social structures by using network and graph theories [13]. SNA has been used in different domains to discover communication patterns among team members. Cross et al. [14] used social networks to visualize invisible patterns of interaction on the management level. The study found that applying SNA was effective in promoting collaboration and knowledge sharing and revealing breakdowns in informal networks. The authors concluded that SNA plays a powerful role in assessing informal structures after a change is done such as acquisition or restructuring.

SNA has also been used by researchers in the information technology field as a tool that gives insight into a software team's communication patterns to aid tackling, among other, communication challenges. Examples of which are the studies by Damian et al [15], which mines task-based social networks to explore collaboration in software teams, and Kwan et al. [16], which aims at answering whether socio-technical congruence has an effect on software build success.

Because of the promising results that were found in using social network analysis in different domains, this project studies the feasibility of automatically constructing social networks using data extracted from software repositories and the possible effect of using them in order to facilitate the collaboration among team members from different departments in Volvo GTT, the industry partner of the study. The social networks in our context are graphs that link owners of system engineering components together based on the system engineering architectural models available in the used repositories.

1.1 Case Company

Volvo Group is a Swedish multinational automotive company. It is one of the world's leading manufacturer of trucks, buses, construction equipment, drive systems for marine and industrial applications [17]. Volvo Group Truck Technology (GTT) is the research and development organization of the Volvo Group. It is a large organization with over 7,000 employees working in global teams. This thesis project is conducted in collaboration with the Electrical & Electronics Engineering (E&EE) department, which creates electronic platforms and is responsible for systems engineering at Volvo GTT. The E&EE department at Volvo GTT uses the tool SystemWeaver [18] as a systems engineering environment. They receive high-level requirements for each project from the product planning, which is outside the department. These requirements are then broken down within the department into smaller parts and assigned to employees, who in turn break down the requirements into logical components. These are on a very detailed level. The resulting component specifications are then handed over to in-house development or used as a contracting document for external suppliers. In parallel to development, the department's testing and verification organization starts to prepare the verification activities independently of the software's source code. The overall product specification is usually maintained and evolved throughout projects rather than written from scratch. SystemWeaver is used for storing the specifications, design architectures, and test specifications.

1.2 Purpose of the Study

The purpose of this thesis project is to facilitate and improve collaborations among team members at Volvo GTT. It aims at developing an approach for using social network originating from structured requirements data to address communication challenges in RE, and evaluating the developed approach in Volvo GTT. The following aims will be targeted during the course of the thesis project:

- Extract use cases for social network analysis in the automotive industry to tackle organizational and communicational RE problems.
- Create a tool to automatically construct social networks based on data extracted from the systems engineering tool SystemWeaver.
- Evaluate the tool at Volvo GTT to discover the possibilities and limitations of the approach.

1.3 Research Questions

In this thesis, we will aim at answering the following three research questions.

RQ1: What are the requirements and limitations for automatically constructing social networks from model-based systems engineering data?

RQ2: To what extent do these networks aid in tackling known communication challenges in RE?

RQ3: Which use cases of this approach are deemed useful by practitioners?

In RQ1, we will look at SystemWeaver, the systems engineering tool used at Volvo GTT, and investigate the feasibility of constructing and using social networks based on the existing data at Volvo. We do that by implementing an artifact that constructs these networks, and retrospectively identify requirements and limitations of the approach.

In RQ2, we will mainly be studying the organizational and communicational requirement engineering challenges that were discovered by other researchers and using the the evaluation of the artifact built in RQ1 to answer whether this technique can be employed to solve those problems and to what extent.

RQ3 aims at evaluating use cases in addition to the suggested aid in tackling communication challenges. This includes additional use cases suggested by practitioners and those proposed by us based on the evaluation of the artifact at the case company.

1.4 Significance of the Study

The findings of this study will have both academic and industrial impacts. In terms of academia, the study addresses challenges which have previously been identified by empirical studies [4, 7–9] using SNA, a technique that has not been used before in the same context to address these specific challenges. The study explores use cases, limitations, and possibilities of the approach, and provides suggestions for conducting similar studies in the future. In industry, Volvo GTT will be the biggest beneficiary of this study, especially that the targeted challenges are directly related to industry. In case of success, the study will serve as a proposed solutions that can be transformed into a professional tool to be used by the organization. In case of failure, the study will give guidelines and suggestions for Volvo GTT to overcome possible reasons for that failure, so that lessons can be learned for future projects. Other companies can also use the findings of this study to replicate it in their own environments.

2

Review Of Literature

In the first two sections, we introduce studies conducted by other researchers that are related to this thesis project. First we present a number of studies that target requirements engineering challenges in the automotive industry, followed by a section to introduce studies that focus on communication and coordination challenges in requirements engineering. In the following sections, we give examples of how social network analysis approach was employed by researchers in different domains, and we finish by highlighting studies where the approach is used in software engineering.

2.1 RE Challenges in Automotive Industry

Almefelt et al. [8] conducted an empirical study aiming to explore experiences and gain knowledge regarding management of requirements engineering in the automotive industry. The study was carried out by conducting 24 interviews with 25 interviewees with different areas of expertise and roles, e.g. concept engineers, development engineers, requirement engineers, manufacturing engineers, and system engineers. Almefelt el al. state that in practice, requirements are often incomplete and conflicting, especially in large scale and complex development environment, which is the case in the automotive industry. Additionally, the authors state that requirements change throughout the projects life cycle, and that it is important to be open to the changes. As a result of the study, Almefelt et al. provide a set of recommendations, one of which is to consider a cross-disciplinary team for the development activities, in order to gain an effective collaboration in the project work. In this project we aim at facilitating the collaboration among project teams, by providing cross-organizational networks of people in a specific context, which aligns with the recommendation provided by Almefelt et al. The intended work also aims to facilitate changing the requirements by providing a set of people who's work might be affected by the change.

Liebel et al. [19] identify seven key problems with communication and organization in automotive RE. Three of these, 'Ch3: Sufficient Context Knowledge', to have an overview over the surroundings on lower levels; 'Ch4: Established Communication and Feedback Channels', to communicate sufficiently with other people within or across the organization; 'Ch6: Clear Responsibilities and Borders', to have clear and communicated responsibilities between different parts of the organization, are essentially related to a lack of knowledge regarding the 'right' persons to talk to in a company. These problems, we believe, could be tackled - or at least reduced - using social network analysis based on data which already exists in many companies - namely the ownerships of requirements and the relations or traces between these requirements.

In an interview-based case study done by Soltani et al. [12], the authors explore the cross-organizational challenges of requirements engineering in the AUTOSAR Ecosystem. The scope of the study targets AUTOSAR-Tier-2 suppliers who deliver software components to AUTOSAR-Tier-1 suppliers who produce Electronic Control Units (ECUs) specifically for an Original Equipment Manufacturer (OEM), and sell them directly to the OEM. The researchers aimed to understand the Requirements Engineering (RE) process of an AUTOSAR-Tier-2 supplier, as well as focusing on what cross-organizational challenges exist in their RE process. As a result they identify that the challenges exist when the AUTOSAR-Tier-2 supplier operates in differentiating segment of the ecosystem, i.e developing specialized components. The researchers found two classes of challenges for the AUTOSAR-Tier-2 supplier in the AUTOSAR ecosystem, which are: Requirements Communication and Requirements Verification. To overcome the challenges the researchers suggest using proven guidelines and tools as well as establishing continuous integration and delivery.

2.2 Communication in RE

Albein et al. [20] report from a literature review that user-developer communication can increase the probability of system success,The authors believe that the requirements engineering (RE) step (requirements elicitation and specification) in software development is a critical one for user participation, as in this step a lot of implicit decisions are taken, some of which should be communicated to the end user. The study aims to create a method to enhance communication between users and developers during RE step. The authors point out that most methods focus on how to get information for requirements from users, but not on how and when to communicate changes in requirements (or in their technical realization) if they are transformed during development. The authors also argue that agile approaches implicitly use that sort of communication, as they claim very close cooperation, they do not work well in large IT projects, as the end user is not constantly on site. The paper references a study [21] on the communication gaps in large IT projects [21]. [21] found out that such gaps are caused, beside others, by complex products, large organization and an unclear decision structure. They identified different effects of missing communication: unmet customer expectations, low motivation to contribute to the requirements work, and developers controlling what is implemented. The authors identified trigger points (i.e.changes on initial user requirements), and the granularity level on which to communicate with the end users. Also, representations of changes and adequate means of communication are discussed. The authors identified trigger points based on the Task Oriented Requirements Engineering (TORE) abstraction levels [22]. They assume most discussions are performed on the domain level. In terms of representation, they suggest the reuse of existing documentation. To find the most adequate means of communication, they look into the media richness theory (MRT) i.e face-to-face and videoconferencing channels, which suggest using rich data channels in case of high equivocal content.

2.3 SNA in practice

Lin et al. [23] study the challenges and solutions in mining and analyzing social networks in enterprise. The authors describe a comprehensive study on the challenges and solutions of mining and analyzing existing social networks in enterprise. Several aspects were considered, including system issues; people's behavior modeling including channel, culture, and social inference; social network visualization in large-scale organization; and graph query and mining. The authors used multiple sources of social data, e.g. emails, instant messages, calendar meetings, and file sharing. The authors describe multiple challenges that were found in the study. For instance they describe a gap caused by not collecting teleconference data and face-to-face interaction data.

Another study conducted by Cross et al [14] discuss that people rely heavily on their social relationships in order to solve problem. The authors state that *both practical experience and scholarly research indicate significant difficulty in getting people with different expertise, backgrounds, and problem-solving styles to effectively integrate their unique perspectives.* The authors approach this problem by using social network analysis with the intention to restructure the organisation to influence on the pattern of informal networks via formal structure.
While this study focuses on building social networks based on multiple sources, it does not consider extracting structured data from data repositories and build networks based on correlations among these data items, which is the case in this project.

2.4 SNA in SE

Damian et al. [24] introduce requirement driven collaboration as collaboration of a cross-functional team of business analysts, designers, developers and testers during the development and management of requirements. The authors describe an approach that (1) constructs a requirement-centric social network which represents the membership and relationships among members working on a requirement and its associated downstream artifacts and (2) outlines a number of social network analysis techniques to study collaboration aspects such as communication. They demonstrate their approach by discussing a case study that examines requirements-driven collaboration within an industrial, globally-distributed software team. Finally, they discuss implications regarding the use of our requirements-driven collaboration approach for research and practice. Damian et al. [15] study Mining Task-Based Social Networks to Explore Collaboration in Software Teams. The study demonstrates the analysis using a social network of failure builds. In the study, the authors describe how current and timely knowledge of the project team's social network is important in many situations, not just with broken builds. One can not always tell exactly who the project experts and central communicators are in the development environment [15]. The study also describes that the importance of the social network is for different roles in the project, i.e. project managers, team leader, and developers. The study states that the team leader can, by examining the team's social network, identify collaboration and communication problems and project newcomers can identify the experts and active communicators. The study also mentions that the use of social networks in software engineering is relatively unexplored and holds much promise for future applications.

Most of these studies do not automate the creation of social networks, but rather create networks manually based on extracted data, in order to perform analysis on them.

3

SystemWeaver

In this chapter, we introduce SystemWeaver [18], the systems engineering platform that we use to conduct this thesis project. We also introduce important concepts that we use in the project.

3.1 SystemWeaver

SystemWeaver is an information management solution for systems engineering and software development [18]. SystemWeaver is developed by Systemite AB [25], a Swedish software development company located in Gothenburg. Systemite developed SystemWeaver with the mission to help its customers manage their information in a way that: increases quality of their products and processes , reduce their development cost, and reduce their time to market. Systemite realized the problem of having multiple sources for a system description e.g. documents, design tools, and hardware tools. Such a fragmented system description implies duplication of information and low tractability and leads to quality issues, inconsistency, and efficiency problems. To tackle this problem, Systemite developed SystemWeaver as a solution that allows users to build a single model that encapsulates the system description, and provides many views on that model.

3.1.1 SystemWeaver Architecture

"SystemWeaver is a model driven development environment where data is governed by a strong meta model that stipulates how models may be built" [18], meaning that the users cannot build anything that is not explicitly allowed by the meta model. Figure 3.1 represents the meta-model of SystemWeaver's conceptual architecture. The meta-model consists of the concepts: item, part, generic node, object, and attribute. The item is the smallest reusable object in SystemWeaver, e.g. requirement, component, list, etc. The part defines a piece of the content of an Item, i.e. a connection between two items. The

attribute is a typed value for an object, this value is unique for the object and can't be used or shared by other objects, e.g. ID, name, description, etc. The node builds a structure to create references to specific instances in an item structure [26].

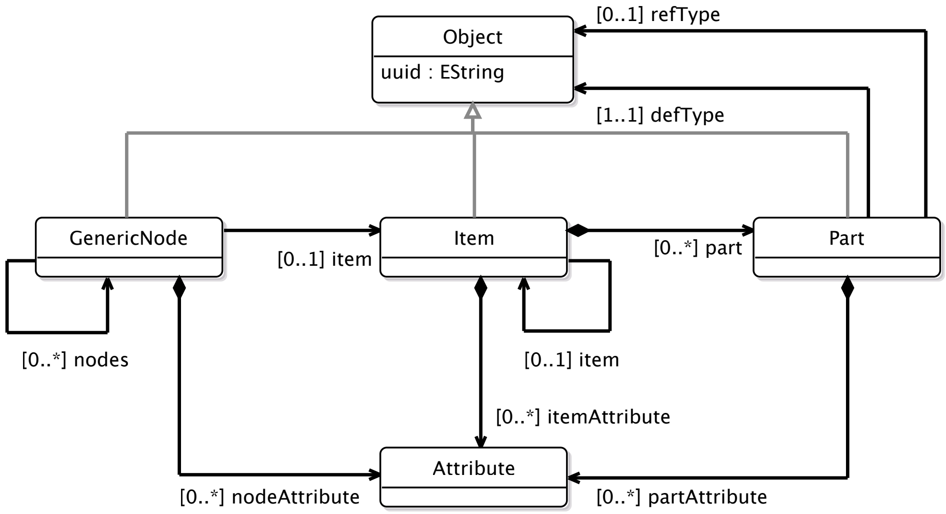


Figure 3.1: SystemWeaver meta-model of the conceptual architecture
Source: SystemWeaver help documentation

3.1.2 SystemWeaver Modules

SystemWeaver contains several modules supporting an integrated embedded system development process. Figure 3.2 illustrates how the modules are applied in four different levels of the system development process. These are: product features, analysis, design, and implementation. Examples of the modules are:

- Feature model: Manages information about the features that a product supports.
- Requirement management: Manages activities related to the products requirement, e.g. authoring, tracing, importing and exporting to other formats, version management, etc.
- Design architecture: Provides tools to manage the components of the product, e.g. component oriented design, allocation, graphical representations, defining interfaces.
- Test and Verification: Provides tools for managing the test and verification process e.g. creating tests from requirements, test execution, requirements coverage, test results and history analysis, etc.

- AUTOSAR: Exports and imports design models into AUTOSAR format, which creates the possibility for collaboration with other environments.

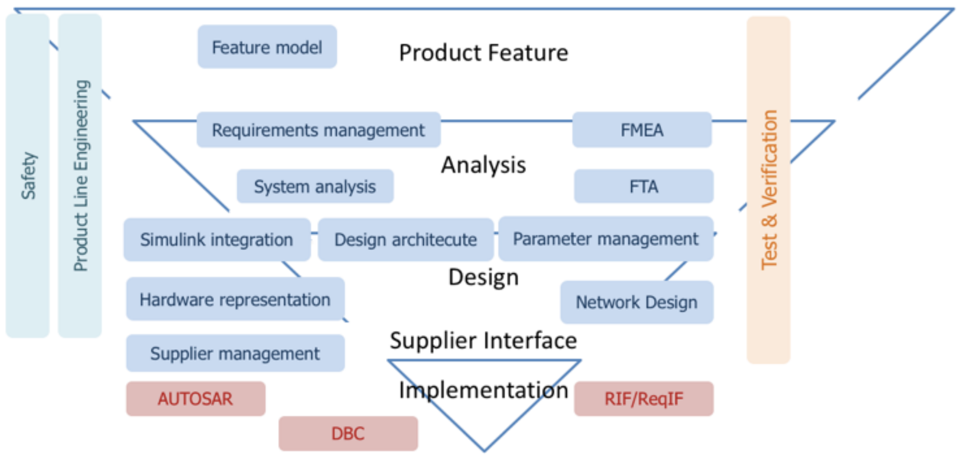


Figure 3.2: SystemWeaver integrated embedded system development process
Source: SystemWeaver help documentation

3.1.3 SystemWeaver at Volvo GTT

Volvo GTT started to use SystemWeaver in 2007 as a platform for system engineering activities, in an evolutionary project called TEA2+, which replaced the prior electric/electronic system. Volvo GTT stores data in SystemWeaver in multiple different abstraction levels which include:

- End to End Functions
- Logical Design
- Hardware Design
- Allocation Design
- Collaborations
- Test and Verification

These abstraction levels are explained in the following six sub-sections.

3.1.3.1 End to End Functions

An End-to-End (E2E) function is an artifact that specifies and represents an end user task. An E2E function includes requirements for this task, and is intended to define a function that transforms a user input to an output back to the user. This means that it represents the functionality that a user experiences in the end product (the truck in Volvo GTT case). Each E2E function has a clear purpose, a defined scope, and at least one involved human actor.

3.1.3.2 Truck Application System

The Truck Application System represents the high-level design structure of the system. It contains Logical Design Architecture (LDA) elements. The functionality of LDAs is defined by Logical Design Components (LDC), which are close to AUTOSAR software components or AUTOSAR software compositions. Each LDC is present only once in the Logical Design perspective (in one LDA), but can however be presented as a part of other LDAs as described in 3.1.1. An LDA also contains the connections between the LDCs as well as presenting in and out ports for signals exchanged with other LDAs.
An LDC may contain many item e.g.:

- Local requirements
- Responsibility requirements
- Receive ports
- Send ports

3.1.3.3 Computation and Signal Distribution System - Node Structure

This abstraction level consists of multiple Hardware Containers (HC), which in turn consist of multiple Electronic Control Units (ECU) or other HCs. The ECUs can be connected to each other by send and receive ports.

3.1.3.4 Installation System - Executable Software System

This abstraction level represents how each of the logical components is allocated and deployed on one of the ECUs. This design consists of Real Allocation Targets (RAT), which links the design to the physical world. More than one RAT may be allocated to the same physical hardware. Logical components are in turn allocated on RAT. One ECU supports one or more RATs. A RAT can be seen as a virtual ECU.

3.1.3.5 Collaborations

Collaborations are sets of LDCs that are connected to each other and interact in specific way in order to accomplish a specific goal. The purpose of a Collaboration is to provide the connection between an End-to-End Function and the design of the functionality in the end product. The Collaboration defines how LDCs interact to realize an End-to-End Function. LDCs are not created in the context of a collaboration, but are rather created in the logical and allocation designs, and used to build up a collaboration. The information in a Collaboration is used as a primary information source from an integration (and verification) point of view. When integrating and verifying the system, the requirements found in the collaboration are used to create the test cases by the testing organization. Collaborations in general contain the following items: included LDCs, signals ,collaboration graph, scenarios, fault situations, signal interaction of interests, and E2E Function reference.

3.1.4 Test Architecture

The testing and verification data is stored according to the meta model represented in Figure 3.3. The figure also shows how the test architecture interacts with the design architecture. The arrows in Figure 3.3 represent parts as described in 3.1.1

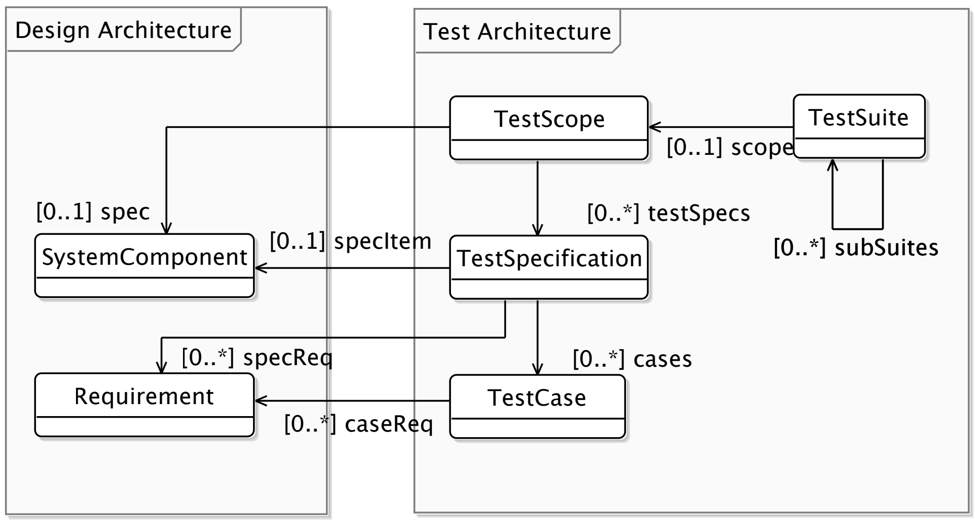


Figure 3.3: Basic Testing and Verification Meta Model

4

Research Methodology

In this thesis project we use the Design Science Research (DSR) methodology. We follow the DSR cycle model described by Vaishnavi and Kuechler [27] and the Guidelines for Design Science in Information Systems Research described by Hevner et al. [28].

4.1 Design Science Research

DSR is fundamentally a problem solving method. It is based on learning about phenomena of interest by constructing and evaluating artifacts. DSR artifacts can be models, methods, constructs, implementations, and design theories [29, 30]. DSR intends to solve previously unsolved real life problems [28], hence the built artifact needs to be innovative and creative. DSR is an iterative method. Within each iteration deeper understanding and knowledge about the problem is built up. The artifact is constructed and improved in each iteration based on the gained knowledge of the problem. At the end of each iteration the resulted artifact is presented to technology-oriented as well as management-oriented stakeholders and evaluated by them. At the end DSR must provide contributions in the area of the design artifact. Table 4.1 contains the seven key guidelines described by Hevner et al. [28], and shows a brief description of each. We followed these guidelines while conducting this project. However in the context of guideline 6, we had little knowledge about the problems and possible uses. Hence, building an artifact was essentially our search process.

4.1.1 DSR vs. Professional Design

One issue that has to be taken into consideration is the difference between DSR and high-quality professional design or system building. Hevner et al. [28] address this issue and clarify that the difference is in the nature of the problems and solutions. On the one hand, professional design applies existing knowledge and technology to solve organizational problems. An example of professional design is the construction of a human resources

Guideline	Description
Guideline 1: Design as an Artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation
Guideline 2: Problem relevance	The objective of design science research is to develop technology-based solutions to important and relevant business problems
Guideline 3: Design evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods
Guideline 4: Research contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies
Guideline 5: Research rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact
Guideline 6: Design as a search process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment
Guideline 7: Communication of research	Design science research must be presented effectively to both technology-oriented and management-oriented audiences

Table 4.1: Design Science Research Guidelines [28]

information system using best practice solutions. DSR on the other hand intends to solve previously unsolved problems in new creative and innovative ways. It is true that building DSR artifacts requires using existing knowledge, but the difference is in the contributions that it offers in the context of its application.

4.2 DSR Iterations

The applied DSR model in this thesis project is the one described by Vaishnavi and Kuechler [27]. The iterative cycle is illustrated in figure 4.1. Each iteration starts with investigating the problem and building awareness of it. Tentative solutions to the problem are then created based on the knowledge gained in the previous step. The tentative solutions are then used to implement an artifact in the development step. The evaluation step starts when the development of the artifact is finished. The last step is

the conclusion which indicates the end of the iteration or the termination of the DSR project. During the suggestion, development, evaluation, and conclusion steps, further knowledge is built up for the awareness of the problem. This extra gained knowledge is used to start the next iterations in the project, if there are any.
We conducted two iterations in this thesis project. The next five sub-sections will introduce in details how this model was used during the project.

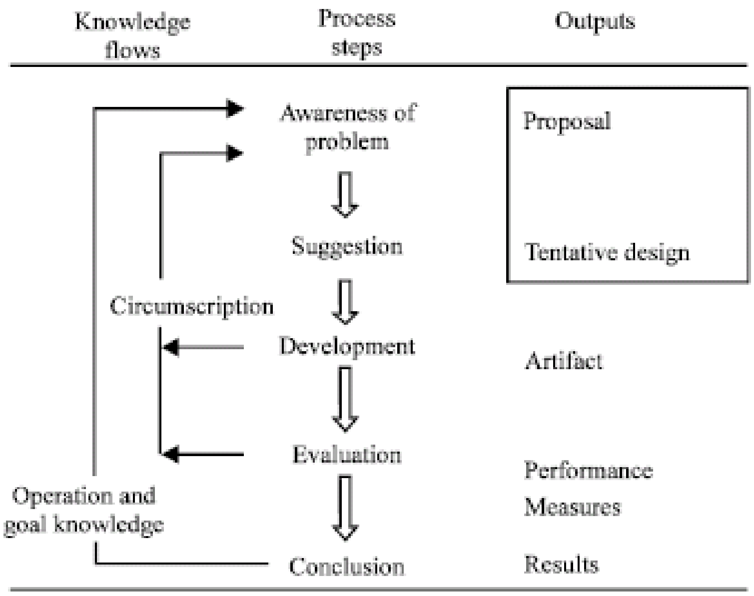


Figure 4.1: The iterative cycle of design science methodology [27]

4.2.1 Awareness of the problem

According to the guidelines in [28], a DSR project targets a solution for a real-life problem. This was the case in this thesis project, which focuses on the requirements engineering challenges in the automotive industry. We started building our awareness of the problem through an exploratory literature review of the the communicational and organizational challenges in the automotive industry. The results collected from the literature review were then projected in the electrical and electronic department in Volvo GTT and were confirmed using unstructured interviews with three practitioners from the department. As this thesis project studies the application of social networks in industry, we conducted a further literature review to discover how these networks have been used in practice. Examples of social network solutions conducted in different industries were studied, and we used the results to build up an understanding of the capabilities and limitations of the approach in general. We proceeded by preliminary identifying communicational challenges that can be tackled using this technology in the context of the project.

We learned more about the problem as the project moved forward. The suggestion, development, and evaluation steps in particular played a crucial role in building up the knowledge and understanding of the problem. The knowledge we gained during the an iteration was used both during the iteration to update and enhance the artifact and in the following iteration (if any).

4.2.2 Suggestion

The suggestion iteration step intends to build a tentative solution based on the findings which emerged from the awareness of the problem step presented in the previous sub-section. The suggestion varied significantly between the two iterations in this thesis project. This was due to the extra gained knowledge in the second iteration, the requested features by the stakeholders in the first evaluation, and the limitation of the used tool in the first iteration.
It was very important to involve stakeholders as much as possible during the suggestion step. This was to avoid suggesting unnecessary features, e.g. a feature that is already provided by the tools used in the department, not to miss important features, prioritize the suggestions according to their importance to the end users to provide the most valuable deliverable within the limited available resources, and to provide early feedback on the applicability of different suggestions. For that sake, we built a focus group consisting of four stakeholders. It consisted of two tool experts, one business expert, and one manager. We consulted this group before each presumably influential decision.
Our basic suggestion was to use the data repository from the requirement management platform used in the EE department to extract available social data and find ways to link the social data together to form a social network. We carefully studied different ways to extract data, identified the social data, considered possible alternatives for the abstraction level, and finally created a suggestion in the first iteration.
In the second iteration, the problem had become clearer but building a tentative solution was more complex because of the deeper and more detailed needs and suggestions gained from the stakeholders that evaluated the first iteration. The focus group played a crucial role at that point as they provided their expertise to carefully analyze the suggestions and build the tentative solution.

4.2.3 Development

In the development phase, the solution suggested in the previous step is implemented. The type of artifact that emerges can vary from one DSR to another. The artifact might be for example an implemented software or a model. In our case the artifact is the implementation of Low-Cost Communication and Coordination (LoCo CoCo), a software tool that automatically creates social networks based on existing systems engineering models.
The development of LoCo CoCo varied significantly between the iterations in terms of complexity, effort, used tools, and time consumption.

In the first iteration, LoCo CoCo was built as a configuration file embedded in a graph library in the used systems engineering platform. We then created social networks of a number of components, resulting in networks in XML format.
We extended LoCo CoCo in the second iteration to become a stand-alone software application that accessed more than one database to create social networks and offered some interactive functionality. This enabled users to alter the visualization of the networks according to their preferences. We further developed features for drawing sub-networks based on multiple criteria, namely a specific person, component, and connection between two nodes. Additionally we implemented features to include people from different departments.
In both iterations, we used the production servers of the case company to create sample networks. We used the latest vehicle platform architecture, containing over 300 main components. In turn, e.g., one of the components we used contains over 150 logical components, over 300 functional requirements, and over 4000 test cases.

4.2.4 Evaluation

The evaluation is a crucial step in DSR. *"The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods"* [28]. The importance of the evaluation step is to asses the work done, as well as to serves as a base for the future work in the iterative cycle of DSR.
The evaluation in each iteration started already during the development phase, conducted in collaboration with the focus group to gain early feedback on the quality of LoCo CoCo and the resulting networks.
In the first iteration, we evaluated LoCo CoCo using four semi-structured interviews. The interviews lasted 45 minutes each. The interviewees were both function owners and function developers. We chose interviews to get rich qualitative data and the interviewees' reasoning around the artifact. The interviews started by an introduction of the study, followed by a presentation of LoCo CoCo. This included a detailed description of how the social networks were built, including examples. We then asked four questions focusing on evaluating the accuracy, limitations, possible use cases, and ways to enhance LoCo CoCo.
In the second iteration, we conducted six in-depth, unstructured interviews. These were conducted with different interviewees than in the first iteration and aimed at having an open discussion about LoCo CoCo. Their primary goal was to asses the usefulness and limitations of LoCo CoCo and gather suggestions on how to enhance it for future work. Additionally, we complemented the qualitative interview data with a survey which we sent to 43 employees, yielding twelve answers from both engineering and management levels from four different departments at the case company. The idea behind using a survey was to involve more stakeholders and to give them the opportunity to freely use the application before submitting their feedback. To do so, we sent out LoCo CoCo and a compact user-guide presentation together with the survey invitation. The survey consisted of one demographic section, one section addressing LoCo CoCo in general, and five sections addressing individual features of LoCo CoCo. To evaluate LoCo CoCo,

we asked closed questions regarding the usefulness and accuracy of LoCo CoCo and its sub-features. Furthermore, free-text questions allowed participants to leave additional comments.
Table 4.2 shows a summary of the stakeholders that participated in the evaluation of the first iteration, the focus group, and the in-depth interviews of the second iteration's evaluation. We are not able to provide similar information regarding the survey, as the responses were anonymous.

Participant	**Role**	**Participation**
Person A	Function Developer	Iteration 1
Person B	Function Developer	Iteration 1
Person C	Function Owner	Iteration 1
Person D	Function Owner	Iteration 1
Person F	SystemWeaver tool Manager	Focus Group
Person E	SystemWeaver tool expert	Focus Group, Iteration 2
Person G	Team Manager	Focus Group, Iteration 2
Person H	SystemWeaver tool expert	Focus Group, Iteration 2
Person I	Function Owner	Iteration 2
Person J	Function Owner	Iteration 2
Person K	Verification Engineer	Iteration 2

Table 4.2: Evaluation participants

4.2.5 Conclusion

The conclusion is the final step in each iteration cycle. It indicates the end of an iteration or an end to the DSR project. Reaching this step implies a satisfying output, even though there could still be a gap between the artifact and the hypothetical predictions.
After first iteration, we basically gained knowledge and lessons that helped us to better understand the problems scope and significantly improve the design of the artifact for the second iteration.
Concluding the second iteration, we were able to conclude the project by answering our research questions and identifying areas for future work.

4.3 Validity Threats

This section discusses the threats to validity in this project based on the classification scheme provided by Runeson et al. [31].

4.3.1 Construct Validity

In order to avoid misunderstandings and misinterpretations, we presented the purpose of the study and the findings of related work prior to each interview in both iterations. Similarly, the invitation email for the survey contained information about the purpose of the study and a user guide in form of a slide show. However, we cannot ensure that participants read or understood all the information in the email.
We were available on-site at the case company on multiple days every week during the project. This additionally raised awareness about the ongoing work.
The survey included questions about the approach as a whole as well as the different sub-features. As this might have confused single participants, we provided a screen shot for each set of questions to clarify which sub-feature we were evaluating.

4.3.2 Internal Validity

This project investigates social relationships among team members in a real life environment. Hence, the developed artifact produces output that might be sensitive to many people. For example, it shows key persons in a given context where many people are involved, as it provides the connections with different roles including the managers, and in some cases it reveals some isolated nodes. Even though the networks are built according to extracted structured data and not according to anyone's judgment, it is possible that the sensitivity causes bias in the practitioners' feedback. For example, if a participant feels that he/she should be a central node in a given context, but the generated network shows someone else, this might influence the participant's feedback. This sensitivity is a potential threat to validity and cannot be ruled out in our current approach. We plan to investigate this and other ethical implications of LoCo CoCo in the future.

4.3.3 External Validity

We conducted the thesis project in one automotive company, constructing networks from models in the tool SystemWeaver. While we think that the approach can be used in an adapted way at companies which own data models linking multiple organization units, e.g., departments, we cannot guarantee this. For example, technical problems with the respective tools or specific company privacy policies could prevent such an adoption. We are currently planning a replication of this study at several different companies in the Swedish embedded domain to investigate the ability to generalize in more detail.
To improve the external validity, we chose interviewees and survey participants from different departments and roles at the case company. Furthermore, we complemented the purely qualitative interview data by a quantitative survey and thus covered a larger sample.

4.3.4 Reliability

SystemWeaver is ideally suited for LoCo CoCo in the sense that it offers a rich API and graph library. Furthermore, we had direct access to experts from the tool provider during the entire study period. This might not be the case for other researchers aiming to replicate this study, which potentially limits reliability. In order to lower the effort for other researchers and companies to replicate our approach, we are currently working on the definition of more general model transformations with the aim to integrate further tools into LoCo CoCo and facilitate adaptation.

5

Iterations

In this chapter we present and discuss in details how we conducted the two iterations of this project. The evaluation step is presented with more details in Chapter 6.

5.1 First Iteration: The Social Networks Characteristics

The purpose of the first iteration was to determine the characteristics of the social networks that are going to be built in LoCo CoCo for the next iteration. These include the data to be extracted, the abstraction level on which the extracted data will be presented, the social data to be used to construct the nodes of the network for example the name of an item creator or the name of a person that has opened and viewed a certain item, and the method of linking the data together to construct the edges of the network. To be able to achieve the aim of this iteration, two approaches were available:

- The first approach was to collect the desired data directly from the stakeholders.
- The second approach was to first build social networks based on initial characteristics and then ask the stakeholders to verify them.

We decided select the second approach in order to present stakeholders with a prototype that could guide the data collection and eliminate possible ambiguities regarding the output.

We constructed the social networks based on data extracted from SystemWeaver [18]. We determined the social data and the abstraction level in collaboration with the SystemWeaver experts at Volvo GTT and Systemite and are discussed in 5.1.1. Our built artifact was evaluated by interviewing stakeholders, resulting in a number of suggestions and enhancements, which will be the main input for the second iteration.

5.1.1 Awareness Of the Problem

At the beginning of the project, we conducted a literature review on Requirement Engineering (RE) communicational and organizational challenges. The literature review revealed a number of challenges, e.g. communicating requirements [21, 32], gaps between requirements and artifacts later in the development process caused by geographic or socio-cultural distribution causes [21], complicate communication caused by distances between teams and engineers [33], and inefficient communication due to lack of awareness between different stakeholders [34].
In a brief informal interview, a manager at Volvo GTT confirmed that the department indeed faces the named challenges.

5.1.2 Suggestion

As discussed in 5.1 the suggested solution in the first iteration was to build social networks based on data extracted from SystemWeaver [18], and evaluate the network's characteristics in collaboration with the stakeholders by interviewing them. In order to build the networks, we had to study four main things, which where the data extraction method, the available social data in the system, what data to be represented as the nodes of the networks, and the connections among the data sets which we will use to represent the edges of the networks.

5.1.2.1 Data extraction

We conducted an unstructured interview with a SystemWeaver tool expert to get an understanding of different possible ways to extract the data needed for constructing the social networks from the tool. The expert explained that there are two ways to extract the data for the purpose of this project.

1. Extracting the data using the SystemWeaver script language.

   SystemWeaver script language is an XML-based language designed to be used by built-in components such as a report generator and a graph generator to extract information from the system. The scripts define what and how the information is presented. This approach works well in order to achieve the desired output for the first iteration, which is the social network. On the one hand, the advantage of this approach is that it does not need any third party tool to work, as the script language is embedded into SystemWeaver as is the graph library which can be used to visualize the network. Another advantage is that the data extracted using the script language has an XML format which makes it possible to be used in the future iteration regardless of the used tool. On the other hand, the disadvantage of using the script language is that its only source of documentation is the SystemWeaver help repository, which is a good source of information, but gives a limited amount of examples.

2. Extracting the data using the SystemWeaver API.

Systemite provides an Application Programming Interface (API) to navigate the data using `C#`. The API is a very powerful approach to extract the data as it provides many possibilities for the users, such as selecting the outcome structure of the data, and selecting the format to present the data. This data extraction approach implies creating a standalone artifact that is not embedded into SystemWeaver. The embedded graph library can however still be used to plot the constructed networks, if we transform them into the XML format used by the library. Other options to plot the graph would be either by implementing our own graph visualization library and embed it into the artifact, or by using an open source graph library that works in `C#`.

5.1.2.2 The Available Social Data

As we intend to build networks that are based on social data, it was important to identify the social data that SystemWeaver includes. We examined the data in SystemWeaver for this purpose, and were able to identify two sources of social data which are

1. A property in each item representing the user that created the item
2. A property in each item representing the last user that committed a change to the item

5.1.2.3 The Nodes Of The Networks

In order to be able to determine the components that are going to be represented as the nodes of the network, we conducted a brief interview with an expert of SystemWeaver tool. The expert responded that the Logical Design Components (LDCs) are the central work parts for the stakeholders of this project, and that it would make sense to have them as the desired abstraction level to build the network. The LDCs are aggregated in different main components, three of which are:

- The Real Allocation Targets (RAT)
- The Logical Design Architecture (LDA)
- The Collaboration

Another item type which can be a candidate to include in the network as per the expert is the ”issue” item type. An issue is a change request to a specific LDC.

5.1.2.4 The Edges Of The Networks

The same brief unstructured interview that was conducted with SystemWeaver [18] expert to determine the nodes of the network was used to determine the edges of the network. The expert mentioned that there are two types of connections for the LDCs which are:

1. Implicit Connections: Each LDC has input and output signals. Two LDCs are implicitly connected if they share the same signal as input to one and output the the other one.

2. Explicit Connection: Two LDCs are connected explicitly using a local connector item which connects the LDCs using their input and output signals.

Fig. 5.1 illustrates the difference between implicit and explicit connections in SystemWeaver. LDC1 and LDC2 are explicitly connected via the local connector Z, whereas LDC3 and LDC4 and implicitly connected by sharing the signal M as input to one and output to the other.

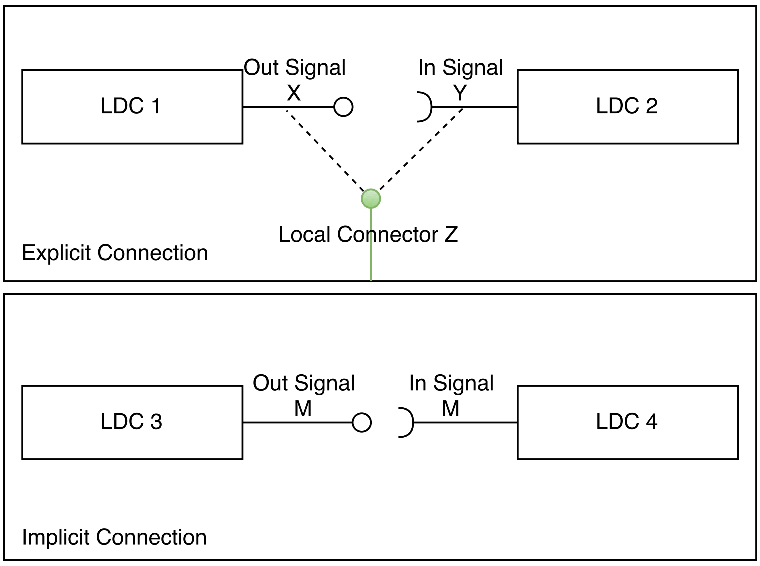


Figure 5.1: Example of implicit connections in SystemWeaver.

5.1.3 Development

To build the social networks; we examined the alternatives presented in Section 5.1.2 and decided that we:

- Extract the data from SystemWeaver using the script language, which satisfies the needs for building the social networks for the first iteration.
- Select LDCs to be nodes of the network as recommended by the interviewee in 5.1.2.3. Further we decided to configure the networks for the three components mentioned by the expert
- For each main component, create two configurations to build the networks. The first configuration using the ”created by” property of the LDCs and the second using the ”last changed by” property.

- Use both implicit and explicit LDC connections to link the network nodes together.
- Use SystemWeaver's embedded graph library to plot the networks.

As a result, two graphs per main component were created.

- Graph1: $G_1(V_1,E_1)$ a complete social network for a main component. Where V_1 is the set of nodes representing the LDCs that belong to the main component by their "Created By" person attribute, and E_1 is a set representing the implicit and explicit connections among the LDCs.
- Graph2: $G_2(V_2,E_2)$ a complete social network for a main component. Where V_2 is the set of nodes representing the LDCs that belong to the main component by their "Last Changed By" person attribute, and E_2 is a set representing the implicit and explicit connections among the LDCs.

Fig. 5.2 shows an example of a network created for a RAT component and plotted using the graph library of SystemWeaver.

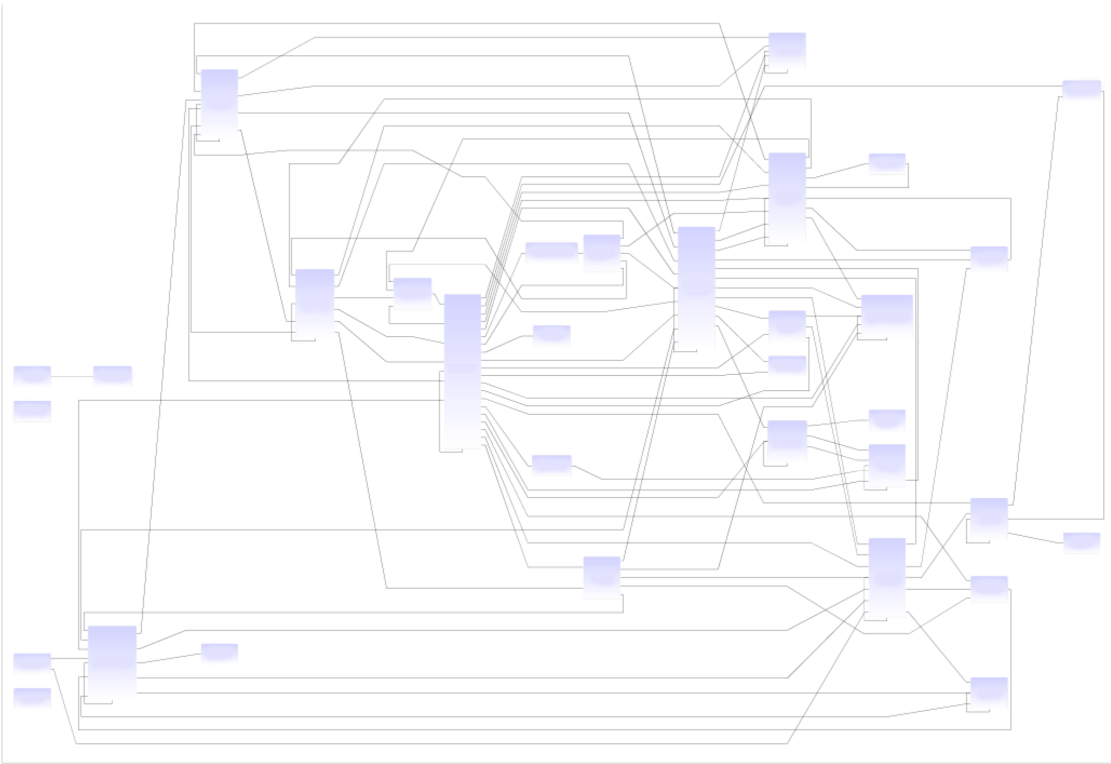


Figure 5.2: Example of a created network in SystemWeaver graph library.

In addition to the complete *main component networks*, a simple application was developed to extract sub-networks for a specific person in the network resulting the following networks for each node:

- Graph3: $G_{vi}(W_{vi},F_{vi})$ a sub-graph of G_1 with the persons node v_i as the root node. the graph has a set of nodes $W_{vi} = \{v \in V(G_1)|\{v,v_i\} \in E(G_1)\}$ that are directly connected to the root node, and a set of edges $F_{vi} = \{e \in E(G_1)|v_i \in e\}$ that is a subset of V_1 edges that connect the nodes in W_{vi} together.

- Graph4: $G_{vi}(M_{vi},N_{vi})$ a sub-graph of G_2 with the persons node v_i as the root node. the graph has a set of nodes $M_{vi} = \{v \in V(G_2)|\{v,v_i\} \in E(G_2)\}$ that are directly connected to the root node, and a set of edges $N_{vi} = \{e \in E(G_2)|v_i \in e\}$ that is a subset of V_2 edges that connect the nodes in W_{vi} together.

Fig. 5.3 shows an example of such a sub-network created for a person in the context of a RAT component and plotted using the graph library of SystemWeaver.

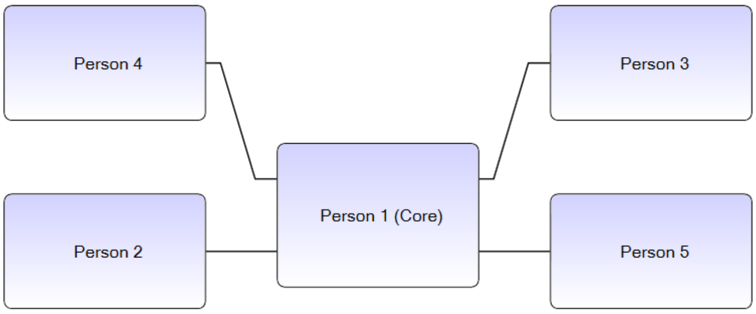


Figure 5.3: Example of a created sub-network in SystemWeaver graph library.

All the graphs were undirected. The reason is that the graphs were intended to show the relation between two persons regardless of the flow of the connection.

5.1.4 Evaluation

To evaluate LoCo CoCo in this iteration, we conducted interviews with four stakeholders. Each interview lasted for 45 minutes starting by an introduction of the project and its objectives, and followed by a set of questions. We presented each interviewee with two complete network of a main component, in which the interviewee was present and had at least four connections. In addition, we presented one sub-network with the interviewees node as the core.
We asked four questions. The first was to evaluate the accuracy of LoCo CoCo. We did this by asking the interviewees to assess to what extent the presented networks reflect their social network in the real work environment. The second question focused on collecting suggestions from the interviewees to enhance the accuracy of LoCo CoCo by either removing or adding connections. In question three, we focused on the potential

usefulness of LoCo CoCo. We wanted to know how much of an opportunity the interviewees see in the social network to help them to overcome some of the challenges discussed in 5.1.1. In the fourth and last question, we asked the interviewees to name use cases where they imagine using LoCo CoCo. We also provided the following suggested use cases for the interviewees to assess:

- S1: Who should I talk to OR who should talk to me
- S2: Who should take over this component
- S3: Who should I send email to and who to copy.
- S4: Who has the most context knowledge
- S5: Who should be in my Skype contact list

The results and discussions of the evaluation are presented in Chapter 6.

5.1.5 Conclusion

As a conclusion of this iteration, we gained knowledge that made us understand the problem significantly better. We also gained a clear idea of how to enhance and extend LoCo CoCo, to better address areas of interest for our stakeholders. In addition, we gained a clear understanding of the strength and weaknesses of the platform which will be a solid ground for us in the second iteration.

5.2 Second Iteration

This iteration focused on enhancing the social network built in the first iteration based on the feedback data collected during the evaluation of the first iteration. We analyzed the suggested enhancements and gave a weight to each of them according to two criteria: Importance and Applicability. We then prioritized the suggestions and implemented the top ranked ones. We extended LoCo CoCo to become a `C#` application that uses data taken from different sources to automatically construct social networks for a context selected by the users. The implementation of this version of LoCo CoCo took five working weeks. During the implementation, we built a focus group consisting of two tool experts, one business expert, and one manager, to give early feedback on the implemented features. At the end of the iteration, we evaluated LoCo CoCo by sending it along with a survey to potential respondents. In addition, we conducted six interviews with people from different departments.

5.2.1 Awareness Of the Problem

During this phase we analyzed the feedback from the first iteration evaluation. There was a clear gap between LoCo CoCo in iteration one, and what the stakeholders considered to be useful. The stakeholders gave a set of suggestions to enhance the quality of the created social networks. The suggestions were categorized in two main categories:

Suggested feature	Importance	Applicability	Weight	Discussion
Add ownership data from a different source	1	0,8	0,8	The stakeholders wanted a more reliable and up to date source of component ownership data.
Add descriptions for the connections	0,75	1	0,75	This suggestion was based on the question ”Why is there a connection between these two nodes”.
Add the implementation engineers to the graph	0,75	0,8	0,6	According to the stakeholders this suggestion would provide data that is ”not easy to find” because they’re in a different department.
Add the owned components names	0,5	1	0,5	This suggestion was based on the question ”Why does this person appear in the graph”.
Add the verification engineers to the graph	0,75	0,5	0,375	According to the stakeholders this suggestion would provide data that is ”not easy to find”. because they’re in different groups in the department.
Add the issues as nodes	0,5	0,2	0,1	The suggestion was to create nodes for the creators of the issues (change requests) and connect them to the corresponding component owners
Add the changers of the component as nodes	0,75	0	0	The suggestion was to create nodes for all the persons that commit changes to a given component, and connect them to the corresponding component owners

Table 5.1: Iteration One Suggested Features

- Network expansion: To include more data, e.g. test cases to the created networks.
- Network visualization: To visualize extra information for the built networks, e.g. what LDCs are owned by a specific person.

We conducted the analysis considering each suggestion separately. We gave an "Importance" rate to each suggestion, derived from the previous iterations evaluation result. There were four function developers and owners in the evaluation, each comprised 0,25 of the total Importance rate, meaning that a suggestion would get for instance an Importance rate of 0,75 if three stakeholders suggested it. We also gave each suggestion an "Applicability" rate, which was determined in collaboration with a focus group consisting of four persons, who had the required knowledge to make a decision about the applicability of a suggestion. We then multiplied the Importance and Applicability to produce a weight for each suggestion, and we prioritized them.

5.2.2 Suggestion

Our suggested solution for this iteration was to create a tool in which the users can select the abstraction level for the network, by selecting a context and a specific component to build the network upon. The available contexts were preserved from the first iteration, and included: Real Allocation Targets, Logical Design Architecture, and Collaboration. The users were to be able to select any component in the contexts. The suggested networks were Logical Design Component (LDC) based as in the first iteration, meaning that the edges would be driven by the connectivity between the LDCs owned by the selected component, and the nodes would be driven by the owners of the LDCs.

5.2.3 Development

In order to implement the set of features that satisfy the needs for a better solution as discussed in 5.2.1, we used a different approach than the one used to implement the first iteration had to be taken. In this phase, we had to implement an external application to be able to access the different data sources needed for constructing the desired social networks.

5.2.3.1 Implemented Features

We implemented six features in LoCo CoCo for this iteration:

Creating a social network based on a user-selected context and component: Each node in the graph corresponds to an owner of at least one LDC, and includes a list of owned LDCs for that owner. We determined the ownership of LDCs in a two steps procedure. The first step was to filter the LDCs according to their existence in the "Owner List", a list of users in SystemWeaver in which each user references the LDCs owner by that particular user. The second step was to assign the owner attribute for the LDCs which were found in the Owner List, and the "Created by" attribute for the LDCs that were not referenced by the Owner List. The owners were distinguished in the application according to the source of the ownership data by applying different styles to different node types. We created the edges of the networks based on the connections between the LDCs and gave a weight that corresponds to the number connection between the edges' nodes. For instance if person1 owns an LDCs that is connected to two LDCs owned by Person2, the edge between Person1 and Person2 would have the weight of two. Figure 5.5 shows an example of an edge weight tool tip. The edges that start and end in the same node were omitted. Figure 5.6 demonstrates how the flow of the network creation process works. We also added a expandable list of the LDCs owned by a person to the persons node. Figure 5.4 shows an example of the list.

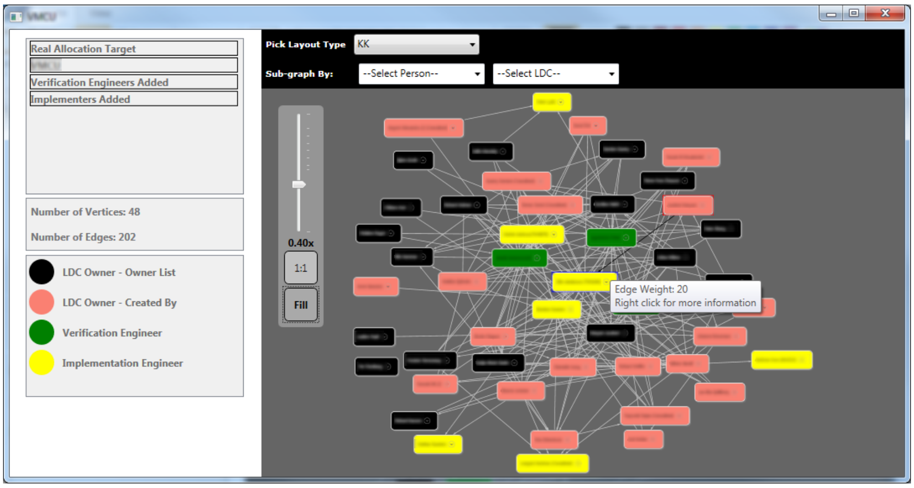


Figure 5.4: Example of an edge weight presentation

Creating a sub-graph based on a specific person: When a graph is created, a list of all the owners (nodes of the graph) was created to enable the user to select one and create a sub-graph with the selected owner as the root node.

Creating a sub-graph based on a specific LDC: When the user creates a graph, a list of all the LDCs that owned by each node in the graph was created to enable the user to select one and create a sub-graph with the selected LDC owner as the root node.

Creating a sub-graph based on a connection between two people: This feature enables the users to view the part of the architectural design that causes the connection of two persons to exist. Figure 5.7 illustrates an example of such sub-graphs
Figure 5.8 shows an example of a sub-network built based on a specific LDC

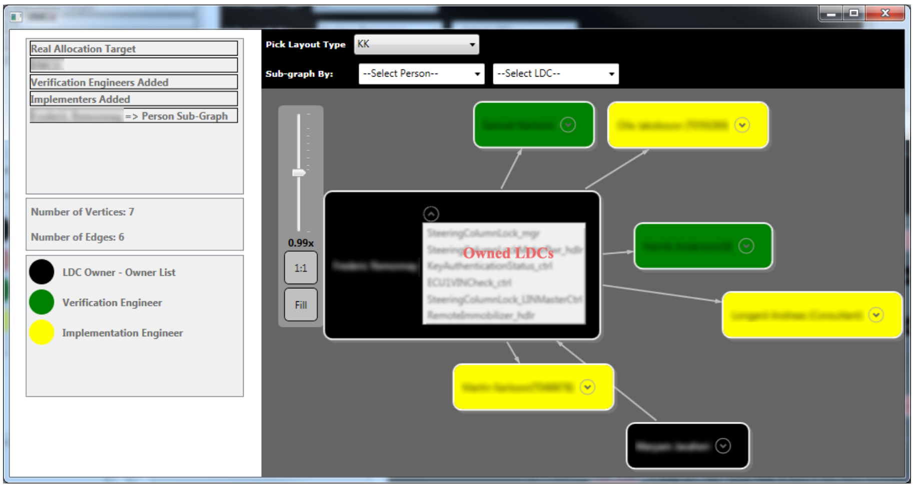


Figure 5.5: Example of the owned LDC list

Adding the implementation engineers: The implementation teams use a database ("implementation database") different from the database which we used in the first iteration ("analysis database"). Accessing and extracting data from the implementation database can be done in the same way as for the analysis database, as it also uses SystemWeaver. It includes data for all the software developed in-house, meaning that implementation data of outsourced development is not available. The databases are connected by linking requirements in the implementation database to requirements in the analysis database. Hence, logical components do not have direct links to the implementation database, but the requirements contained in each logical component has to be exploited instead.
Fig. 5.9 outlines how LoCo CoCo uses these links to find a connection between people in both databases.

Adding the Verification Engineers To add information from verification items to LoCo CoCo, we traverse the items in the test and verification model to find links between the test cases and the logical components of the network. When a match is found, we add the engineer responsible for the test specification that contains the test case to the network, and add a link between that person and the owner of the logical component.
The very large data set and the unidirected nature of the relations between verification items and the logical design makes traversing all the test cases a very time consuming task. Therefore, we decided to only implement this feature for a subset of the components as a pilot.

Additional graph information: In order to better understand the created networks, we provided further information, i.e., the source of the network, the number of nodes and edges, and a legend for the different node types and colors.

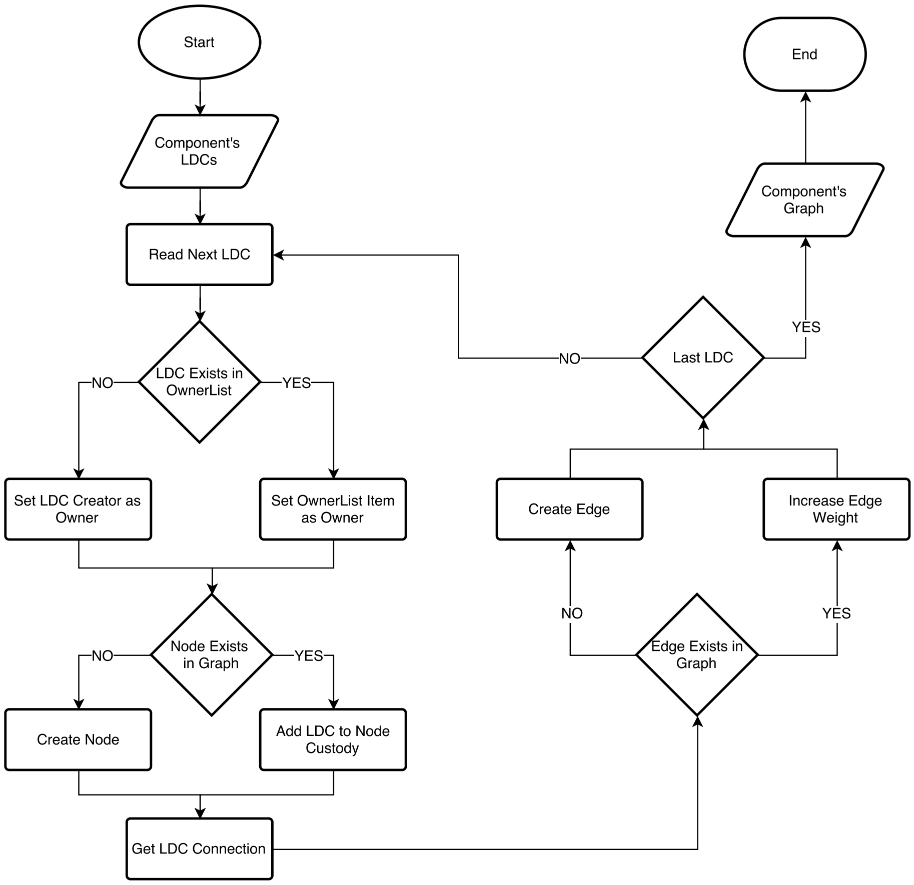


Figure 5.6: LoCo CoCo Network Creation

5.2.4 Evaluation

Our primary goal for this iteration's evaluation was to asses the usefulness and limitations of LoCo CoCo and gather suggestions on how to enhance it for future work, in addition to gathering possible use cases for the approach.
We did this by sending out a survey and conducting unstructured in-depth interviews. We chose the survey to gain feedback from as many stakeholders as possible and to give them the freedom to try out the artifact before submitting their feedback. Whereas in the in-depth interviews, we intended to gain deeper feedback from experts with different roles.
We sent out the survey along with LoCo CoCo and a user guide to 43 employees, yielding twelve answers from both engineering and management levels from different departments at the case company. The survey consisted of one demographic section, one section addressing LoCo CoCo in general, and five sections addressing individual features of LoCo CoCo. We used two types of questions in the survey. These are quantitative questions

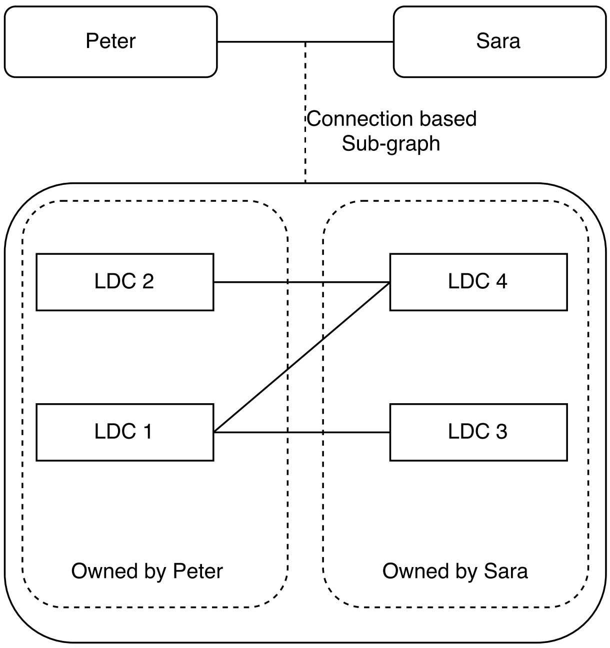


Figure 5.7: Example of a sub-graph based on a connection of two people

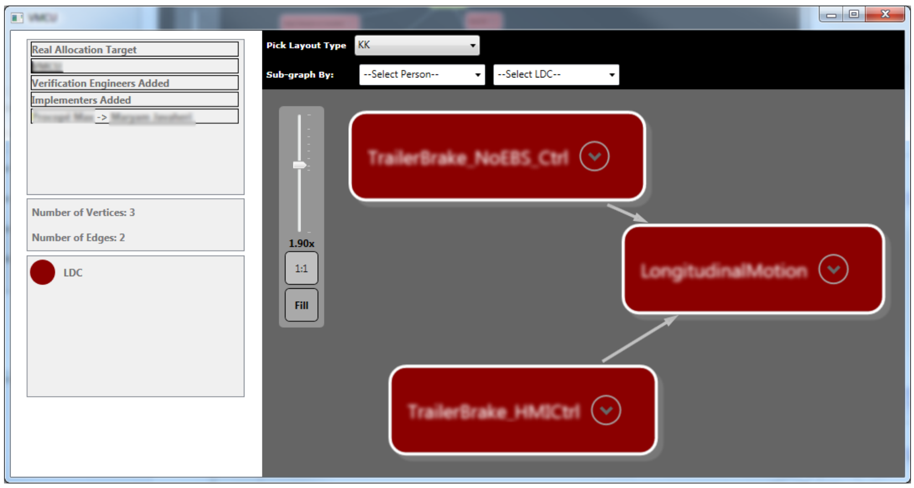


Figure 5.8: Example of a sub-network based on a connection between two persons

were the users assessed the usefulness and accuracy on a 1:10 scale, and qualitative questions, where respondents textually expressed how LoCo CoCo and its features can be improved and used in their everyday work.
We conducted the in-depth interviews with six interviewees different from the one that evaluated LoCo CoCo in the first iteration. We aimed at having an open discussion

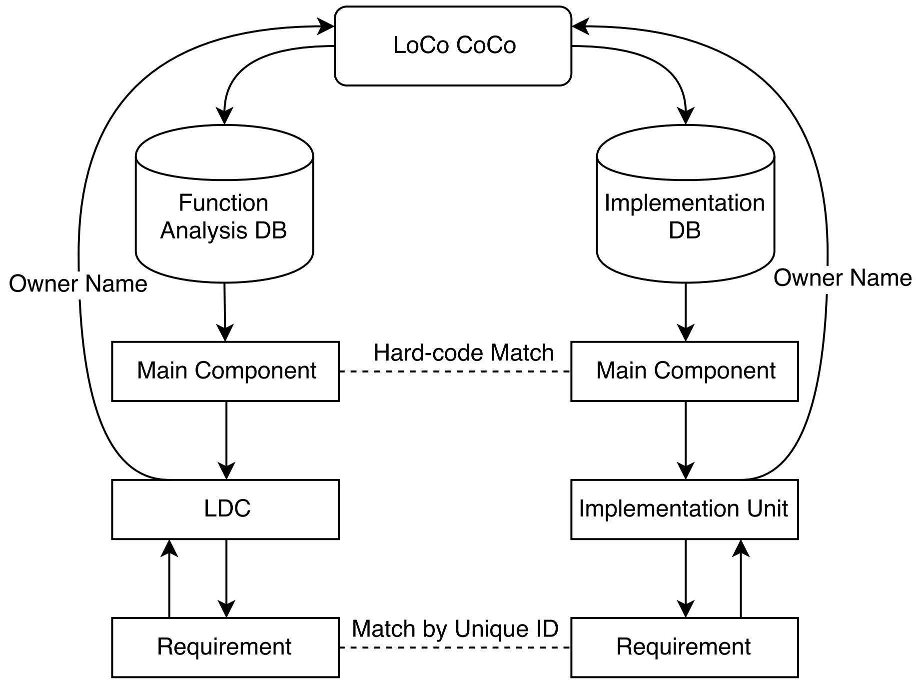


Figure 5.9: Adding Implementation Engineers in LoCo CoCo

about LoCo CoCo.
The results from both the survey and the interviews are presented in 6.2 Additionally the survey questions is presented in Appendix A

5.2.5 Conclusion

As a conclusion for the second iteration, we were able to conclude the project by answering our research questions and identifying areas for future work. Detailed conclusion is discussed in Chapter 7

6

Results and Discussions

In this chapter we present and discuss the evaluation results of LoCo CoCo in both conducted iterations.

6.1 Iteration one results

Accuracy The interviewees gave an average rating of 4 out of 10 for the accuracy of networks based on the "Created By" property, and 4.5 out of 10 for those based on the "Last Changed By" property.
One of the interviewees explained this low rating by stating that "*There are some nodes of people that have left the company, and some for people that have moved inside the company and have new positions that are not related to the context of the network*".
To make the networks better reflect their connections, the interviewees mentioned the following list of data to be added to the network:

- The owners of the components from the owner list to replace the currently used attributes
- The issues as new nodes
- The function owners as new nodes
- An attribute to the components name
- A description of the connection
- The system leaders as nodes
- The implementation team as nodes
- The testers as nodes

- The software owners as nodes
- The hardware owners as nodes
- The persons who committed changes to the item as nodes

At this point we didn't have proper knowledge of all of the suggested data. Hence we decided to keep a record of them to examine and analyze them later in the next iteration. The interviewees also asked to remove the persons that have changed their roles or left the company from the nodes.

Usefulness The interviewees gave an average rating of 2 out of 10 for the usefulness of LoCo CoCo.
When we asked the interviewees to suggest use cases for the approach, we got no answer, as they did not have a clear idea of how the networks might aid them in the context of their work. However when we suggested and provided a number of use cases, they emphasized that it can be used to know who to talk to regarding a specific component, or who the person with the most context knowledge is, especially for new employees.

The results taken from evaluating this iteration show that there is a gap between the networks constructed and the desired networks to solve the problems investigated in 5.1.1. We think that this gap can be due to one or multiple of the following causes:

- Collective edit permission: SystemWeaver adopts the philosophy that anyone with the right role can commit a change to any item in the architecture. This makes assigning custody of an item to a specific person a hard task to accomplish, hence causes potential inaccuracy in the constructed networks.
- Lack of change logs: This possible cause completes the previous one. Having no change log, means that we can not know who has been working on an item, but rather only know the last changer. This information could otherwise be a valuable source of social data.
- Uni-department solution: LoCo CoCo at this point provided networks of people working in the same department. This might have caused the stakeholders to consider it to be trivial. However, expanding the solution into a cross-organizational one might have a good effect on the usefulness of LoCo CoCo
- Uni-platform solution: Similar to the previous potential cause, expanding the solution into a cross-platform one might have a good effect on the usefulness of LoCo CoCo.

6.2 Iteration two results

In this section, we present and discuss the results from both the survey and the in-depth interviews conducted in the evaluation step of the second iteration.

6.2.1 Usefulness of LoCo CoCo

The survey data shows that LoCo CoCo is considered to be useful for most of the participants, as depicted in the box plot in Fig. 6.1. Nine out of twelve participants find the approach useful or somewhat useful.

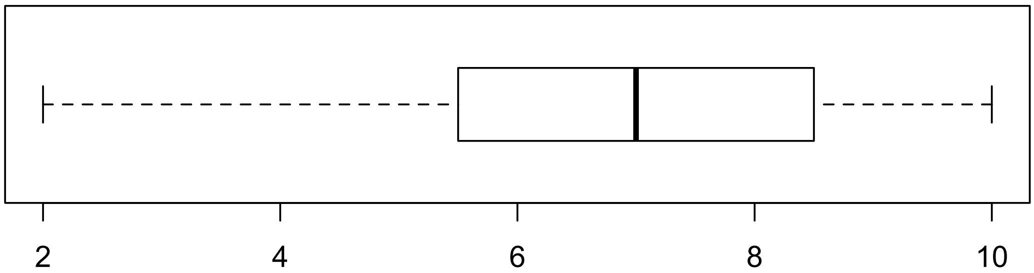


Figure 6.1: LoCo CoCo usefulness

All twelve participants state that they would use LoCo CoCo for at least one of the proposed use cases. Out of those, eight participants stated that they would use the approach for more than one proposed use case. While the "finding the right person to talk to" use case received the highest rating (eight participants), the remaining proposed use cases received nearly the same rate. Both "Finding out who will be affected by a committed change" and "Gathering the required persons for a meeting set-up" were selected by seven participants, and "Finding the persons with most context knowledge" by six.

Three additional use cases were suggested by the participants. These are to find the related contacts when taking over someone's tasks, to overview the correlations among team members for the management level, and to provide an overview of contact persons from different departments for new employees.

With respect to the usefulness of LoCo CoCo, the interview data was aligned with the survey data. All six interviewees felt that LoCo CoCo is useful and that there are multiple use cases that can be extracted from it. Two of the interviewees stated that it provides an easy way to get instant feedback of the complexity of different components. This information is known in most cases in the system architecture itself according to the interviewees, but yet visualizing it from a social perspective provides additional knowledge. In addition, the interviewees felt that LoCo CoCo could provide a good way to visualize vulnerability of communication due to isolated nodes in the generated graphs. These nodes could reveal a potential lack of communication, a lack of connectivity in the architecture, or both. We believe that this lack of connectivity may serve as a trigger for the employees to go back to the systems engineering tool to make changes or to bring these issues up with colleagues.

6.2.2 Limitations of LoCo CoCo

Similar to the first iteration's evaluation discussed in 6.1, accuracy was still mentioned as a limitation in the second iteration's evaluation, even though the usefulness was evaluated much higher. Seven participants stated that the ownership data was not up to date in the tool used to extract data, which led to false information in the generated networks. It is important to point out that this limitation is not related to the way we extract data, but rather to the data quality.

Another limitation according to two survey participants is due to LoCo CoCo being developed standalone. Even though the artifact uses SystemWeaver's database, the participants felt that it should be integrated into SystemWeaver. This limitation only surfaced during the second iteration, as we used the script language and graph library embedded in SystemWeaver during the first iteration. We think that this integration is feasible in the future due to our close collaboration with the tool vendor.

Three participants felt that they had more knowledge of who to contact than what is offered by the tool, and hence felt that it is easier and less time consuming to contact the persons directly. This might be due to the experience of the participants, having worked for a long time in their departments. Therefore, LoCo CoCo could still help engineers newly recruited or switching roles. We think that experienced engineers could still benefit if more data spanning across the organization would be added to LoCo CoCo.

Two participants referred to the fact that not all the features were available for all components, according to them a limitation of the approach. This is however not a limitation of our extraction approach, as the reason for the missing ECUs was due to lack of trace links between the required items in the model.

The in-depth interviews did not reveal more limitations than the ones found in the survey. Yet, we were able to get more understanding of the causes for the limitations. All interviewees noted that the used data was outdated to some extent. One interviewee stated that *"Due to the nature of our work, items are reused rather than recreated, which makes the creator attribute easily outdated when the person leaves the company or switches role for instance"*. In our opinion, this defeats the purpose of having these properties in the items in the first place. The interviewee however revealed that the reason for the ownership data to be outdated in many cases, even in the owner list, is that *"it is considered as a low priority to update this data compared to other core business tasks"*. We believe that LoCo CoCo could trigger employees to update the ownership data.

6.2.3 Evaluation of Features

We evaluated six implemented features in three categories. These are:

- One main feature: creating a network based on a specific content.
- Three sub-Network features: based on a specific person, a logical component, and a connection between two people.

- Two context broadening features: adding implementation and verification engineers.

We implemented this set of features based on the interviewees' feedback in the first iteration and used the usefulness and accuracy as metrics to evaluate them. Fig. 6.2 shows box plots for the usefulness of each of the six features. The answers are on a scale from one to ten, where ten is very useful and one is not useful at all. The scored medians of the features are similar, with the context broadening ones being slightly higher and the person based sub-network being lowest. The interquartile ranges are also similar for most features except for the nodes connections and person based sub-networks being shorter. However, no feature scored a mean rating lower than five, which we believe indicates the potential of the features' usefulness.

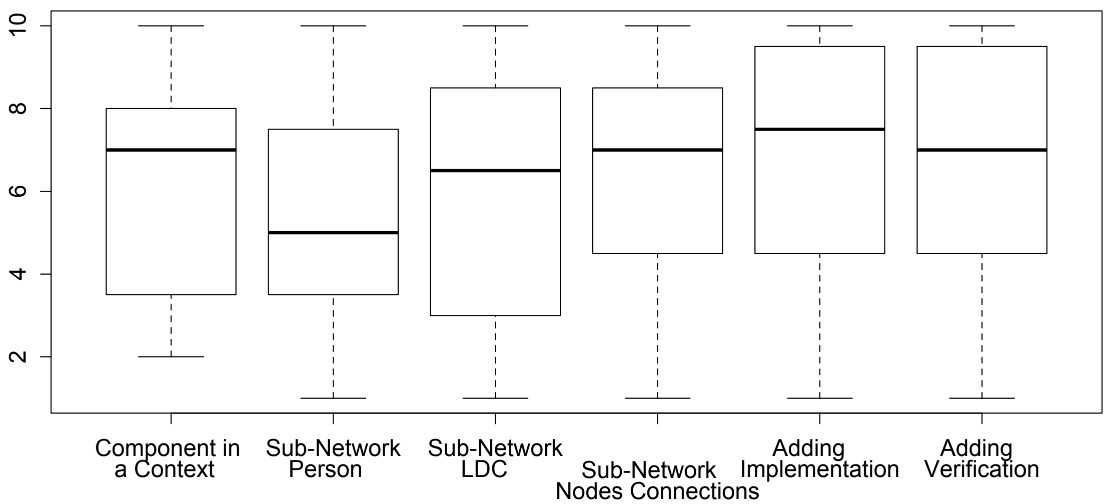


Figure 6.2: Survey data for average usefulness of different features

The accuracy assessment is intended to show how close the built networks are to real life (see Fig. 6.3). A ten-point scale is used where ten is perfectly accurate and one is not accurate at all. Similar to the usefulness, the accuracy median rates for different features are similar to a large extent. The interquartile ranges are also similar except for nodes connection based sub-network and component in a context features being shorter. No feature scored a mean rate less than five, which indicates relevancy between the networks and real-life networks, despite the limitations mentioned previously.
As Fig. 6.2 and 6.3 already indicate, there is a strong correlation between the usefulness and accuracy answers for all six features. These are listed in Table 6.1.
While these correlations could be interpreted in several different ways, based on the interviews and some of the free-text comments in the survey we think the following explanation is most likely. Participants are less critical towards the accuracy of a feature if they value the usefulness of it. Additionally, participants who observe a low accuracy tend to rate the usefulness lower. We are not aware of any external causes that might

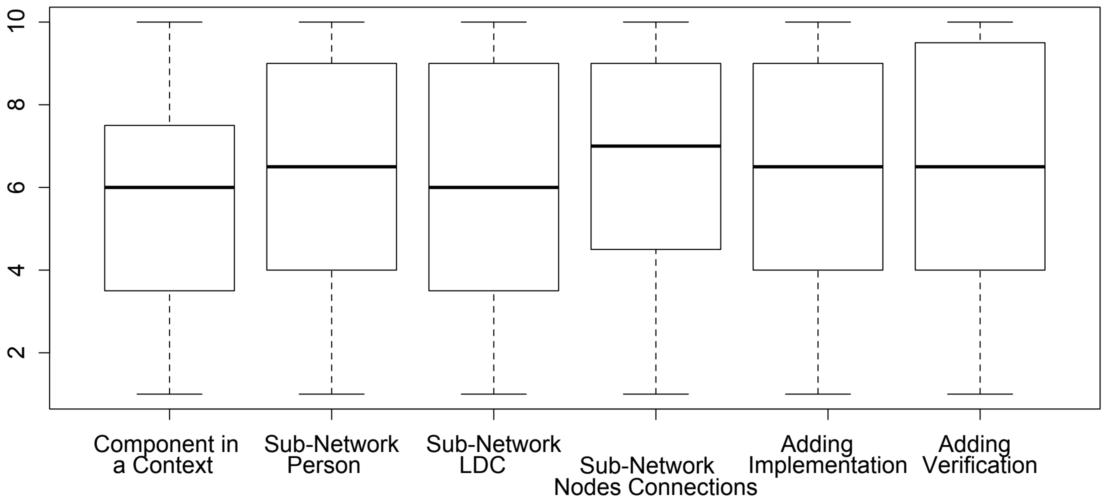


Figure 6.3: Survey data for average accuracy of different features

Feature	**Spearman's ρ**
Component in a Context	**0.7089****
Sub-Network Person	**0.7558****
Sub-Network LDC	**0.9415*****
Sub-Network Nodes Connections	**0.9311*****
Adding Implementation	**0.8434*****
Adding Verification	**0.9214*****

*p<0.05, **p<0.01, ***p<0.001

Table 6.1: Correlations between Usefulness and Accuracy for LoCo CoCo Features

affect this correlation. While they indicate that not all usefulness and accuracy answers are in fact appropriate, they support the observation from both interviews and free-text answers that the accuracy is highly affecting the usefulness of LoCo CoCo.

6.2.4 Enhancement of LoCo CoCo

To make the approach more useful, survey participants suggested a number of enhancements and features to implement. These suggestions can be categorized as follows.

Enhancement of accuracy As discussed earlier, the approach is limited because of the data accuracy. Six participants suggested using a more reliable source of social data.

One participant suggested to use a different approach for finding the owner of a specific component by assigning the custody to the person that made the highest numbers of changes to it. As this data is currently not available, this enhancement would depend on a decision by the case company or tool provider and is out of scope for LoCo CoCo.

Completeness of current features The version of the artifact used in the second iteration allowed adding verification and implementation engineers for a limited number of components. Five of the participating practitioners suggested to offer this feature also for the rest of the components in SystemWeaver's database. Four participants suggested to complement the provided social data with other sources, e.g., the human resource system, to provide more in-detail information about the persons that are included in the graphs. Furthermore, participants felt that the networks' data should be verified against the data from these additional sources. However, accessing data from human resources and accumulating data from multiple systems in general could be prevented by policies or even laws, and might raise ethical issues. We are currently planning a follow-up study investigating these ethical issues.

Add-On features Two participants stated that the approach should be turned around, thus providing the architecture of a component first and then showing the related stakeholders for that component. The reason behind this, according to the participants, is that employees have more knowledge about the architecture compared to the people involved. This could be because of the nature of the platform being a systems engineering tool. However, it defeats the purpose of building social networks and indicates that practitioners at the case company also currently lack a comprehensive overview of the architecture.

In summary, several useful and interesting features were suggested by survey participants and interviewees. However, some of these raise further issues, e.g., ethical problems or technical difficulties, that warrant future investigation.

7

Conclusion and Future Work

In the course of this study, we aimed to enhance the requirement engineering process in the automotive industry by tackling the communicational and organizational RE challenges found by other researchers. Our proposal was to use existing data to build and visualize social networks to facilitate requirements-related communication and coordination in system development.

We conducted a design science study with practitioners and experts, and used real life data on the premises of Volvo GTT. In the context of the study, we developed the approach LoCo CoCo i.e., Low-Cost Communication and Coordination, to automatically build and visualize social networks based on the system engineering data at Volvo GTT.

The development of LoCo CoCo made it possible for us to answer our first research question by retrospectively identify requirements and limitations of LoCo CoCo. The practitioners that evaluated the approach emphasized the usefulness of LoCo CoCo, as it provides instant feedback, identifying proper contact persons in a given context, identifying who will be affected by change, and visualize emergent teams. These use cases answer our second and third research questions.

The evaluation of LoCo CoCo revealed some remaining limitations. These are mainly caused by the low quality of the available social data. Participating practitioners suggested enhancing features of LoCo CoCo and the accuracy of the data to further improve the approach. In addition to that, the practitioners emphasized on the importance of expanding LoCo CoCo to become a cross-organizational approach, in order to enhance its usefulness. To achieve that LoCo CoCo needs to use data from multiple tools used by different departments.

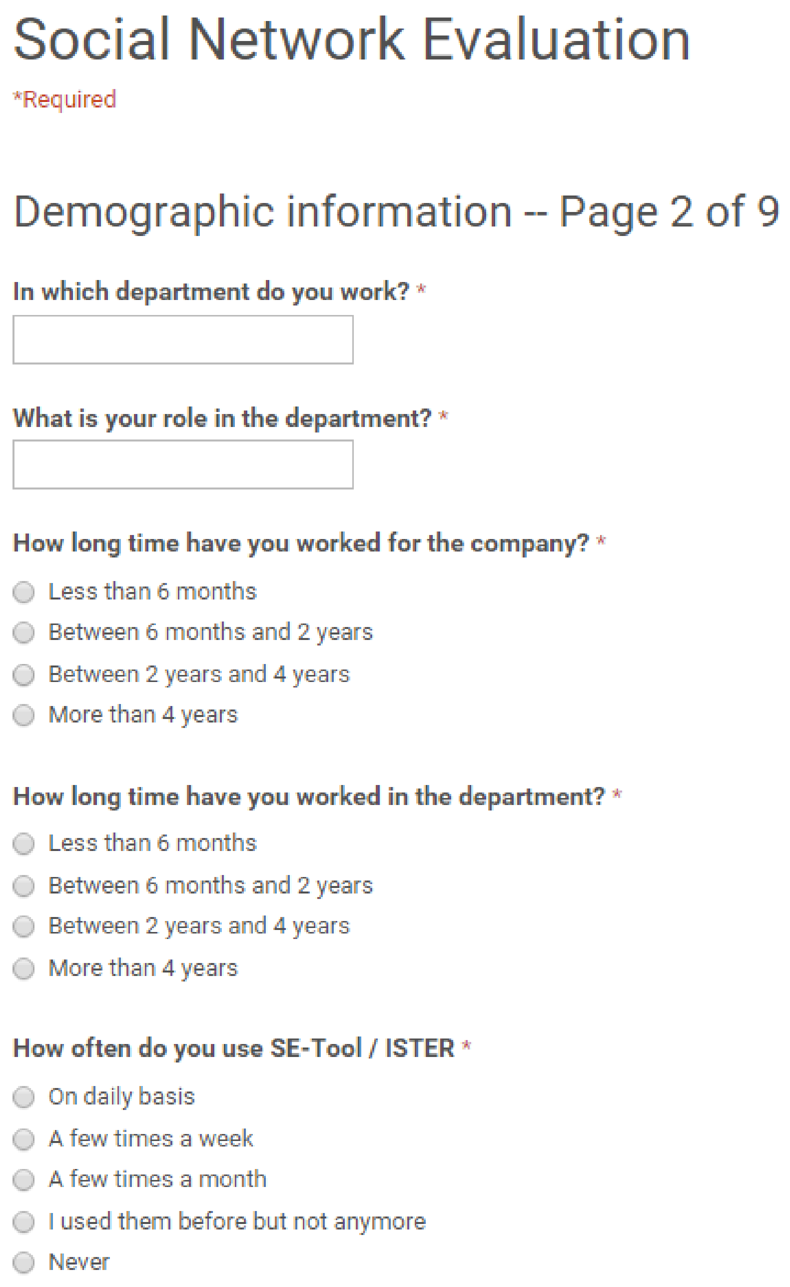
We believe that LoCo CoCo provides additional knowledge, that is sufficient to trigger employees to improve the data quality in the original models, since it provides immediate benefit for this effort. In this way, both the systems engineering models and the social network models would benefit from continuously improving quality.

We would like to encourage further studies to expand upon these socio-technical aspects of system engineering modeling and tooling. In particular, LoCo CoCo should be extended to further existing systems engineering tools and meta models. The implications of generalizing the approach, and spanning different tools and databases, both with respect to technical issues and to organization or ethical issues should be studied in further detail.

8

Appendices

APPENDIX A : Survey questions



Social Network Evaluation

*Required

Demographic information -- Page 2 of 9

In which department do you work? *

What is your role in the department? *

How long time have you worked for the company? *

- Less than 6 months
- Between 6 months and 2 years
- Between 2 years and 4 years
- More than 4 years

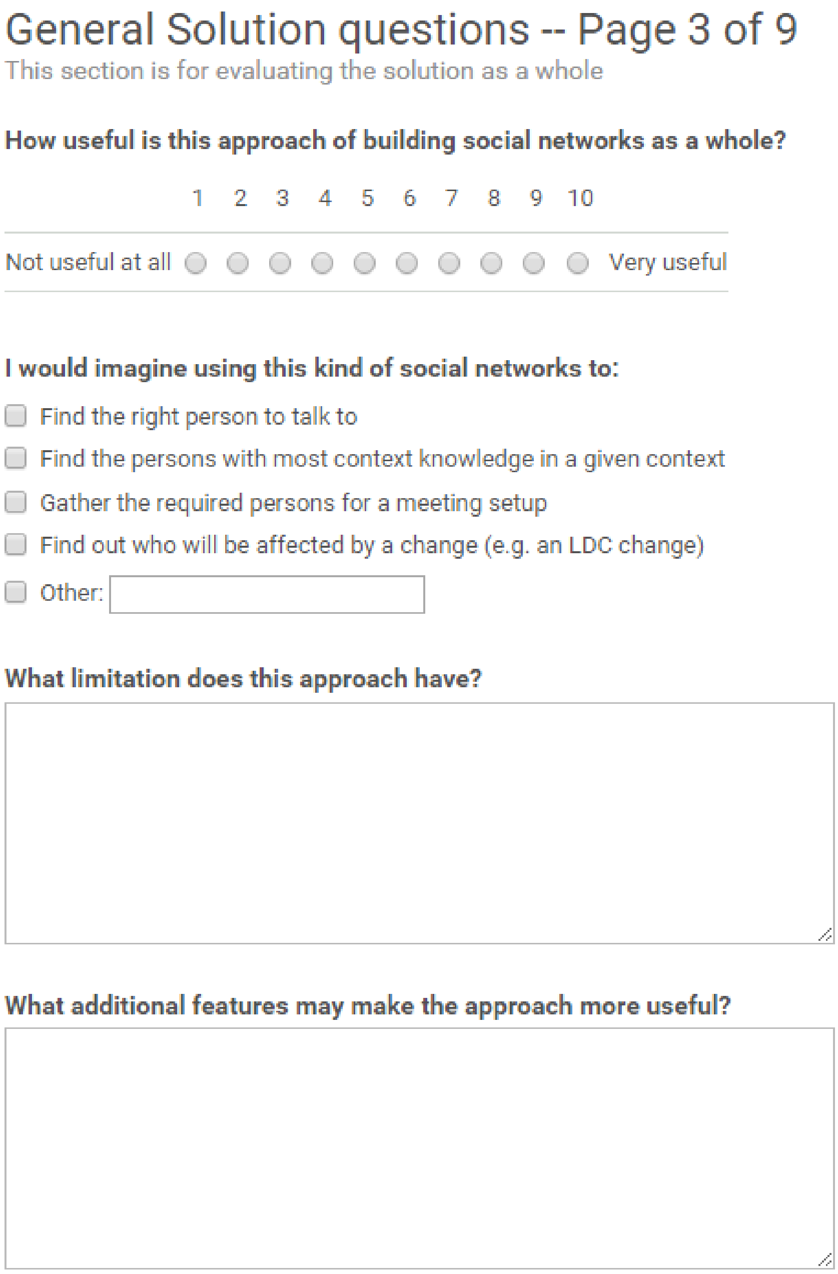
How long time have you worked in the department? *

- Less than 6 months
- Between 6 months and 2 years
- Between 2 years and 4 years
- More than 4 years

How often do you use SE-Tool / ISTER *

- On daily basis
- A few times a week
- A few times a month
- I used them before but not anymore
- Never

Figure 8.1: Second Iteration Survey - Page 2



General Solution questions -- Page 3 of 9

This section is for evaluating the solution as a whole

How useful is this approach of building social networks as a whole?

	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

I would imagine using this kind of social networks to:

- ☐ Find the right person to talk to
- ☐ Find the persons with most context knowledge in a given context
- ☐ Gather the required persons for a meeting setup
- ☐ Find out who will be affected by a change (e.g. an LDC change)
- ☐ Other:

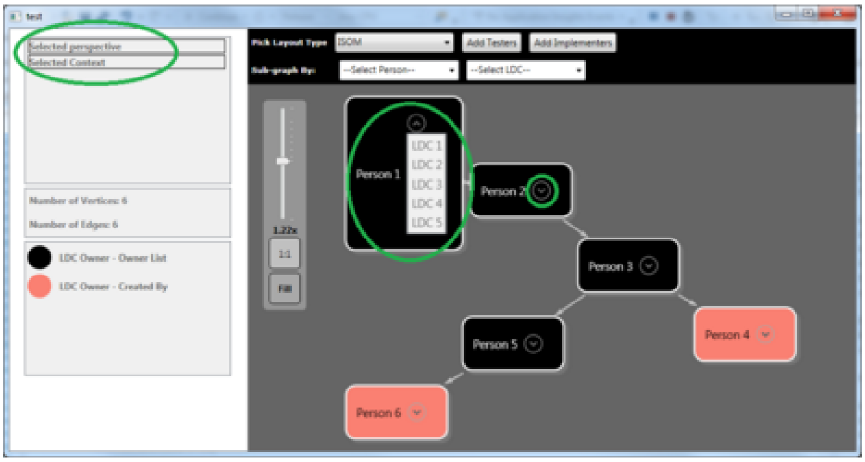
What limitation does this approach have?

What additional features may make the approach more useful?

Figure 8.2: Second Iteration Survey - Page 3

Social network for a component in a context -- Page 4 of 9

By context we mean a Real Allocation Target (RAT), a Logical Design Architecture (LDA), or a Collaboration.



How accurate are the generated graphs for a component in a context?

By accurate we mean how close the generated graph is to the real life social network

	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Perfectly accurate

How useful are the generated graphs for a component in a context?

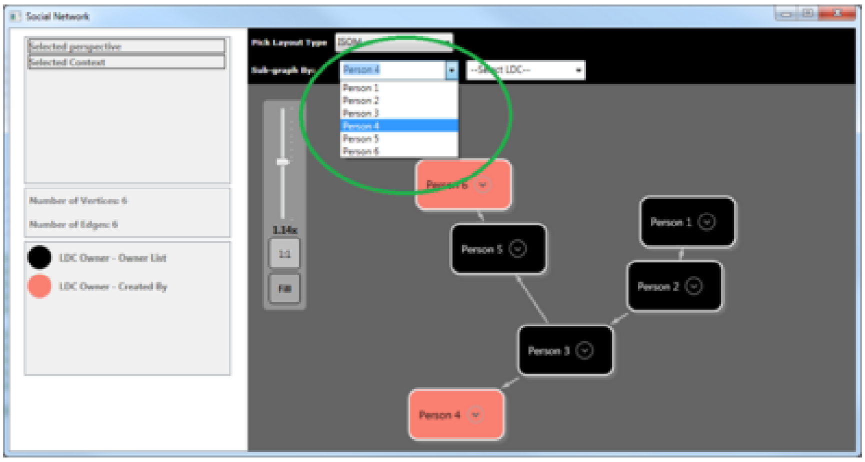
	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

What are the limitations of this feature and how can it be improved?

What possible use cases does this feature offer to your work?

Figure 8.3: Second Iteration Survey - Page 4

Sub-Graph network for a specific person -- Page 5 of 9



How accurate are the generated sub-graphs for a specific person?

By accurate we mean how close the generated graph is to the real life social network

	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Perfectly accurate

How useful are the generated sub-graphs for a specific person?

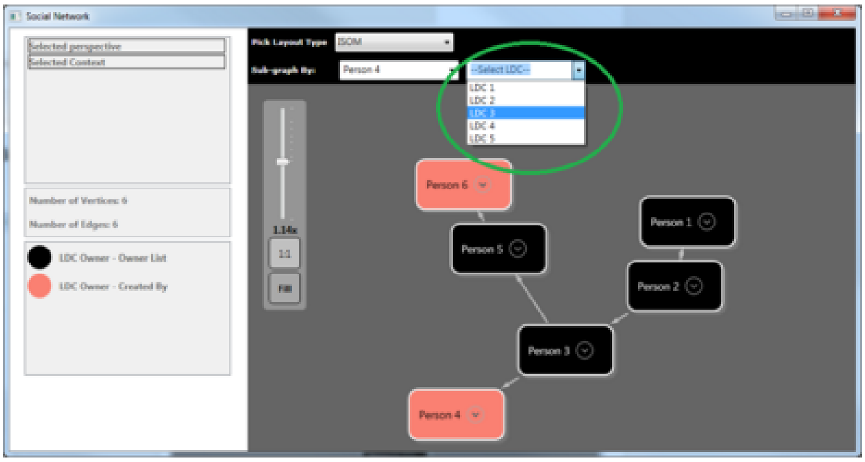
	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

What are the limitations of this feature and how can it be improved?

What possible use cases does this feature offer to your work?

Figure 8.4: Second Iteration Survey - Page 5

Sub-Graph for a specific LDC -- Page 6 of 9



How accurate are the generated sub-graphs for a specific LDC?

By accurate we mean how close the generated graph is to the real life social network

	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Perfectly accurate

How useful are the generated sub-graphs for a specific LDC?

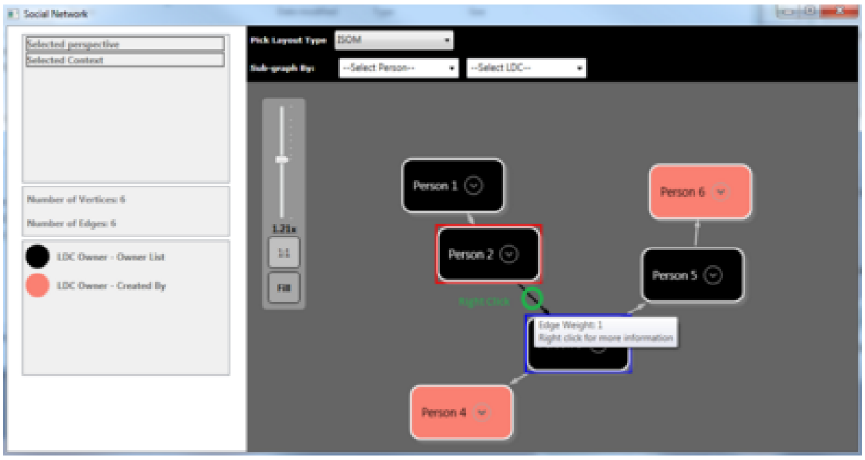
	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

What are the limitations of this feature and how can it be improved?

What possible use cases does this feature offer to your work?

Figure 8.5: Second Iteration Survey - Page 6

Sub-Graph based on a connection between two people -- Page 7 of 9



How accurate are the generated sub-graphs based on a connection between two people?

By accurate we mean how close the generated graph is to the real life social network

	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Perfectly accurate

How useful are the generated sub-graphs based on a connection between two people?

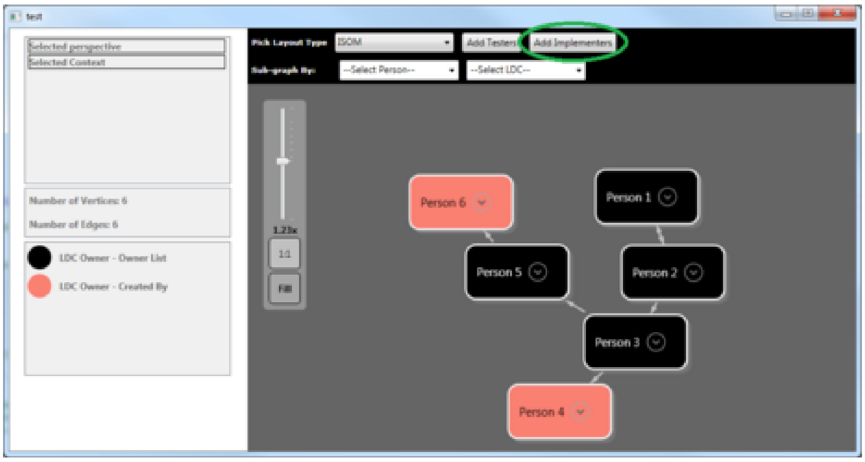
	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

What are the limitations of this feature and how can it be improved?

What possible use cases does this feature offer to your work?

Figure 8.6: Second Iteration Survey - Page 7

Adding the implementation engineers to a graph -- Page 8 of 9



How accurate are the generated graphs after adding the implementation engineers?

By accurate we mean how close the generated graph is to the real life social network

	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Perfectly accurate

How useful are the generated graphs after adding the implementation engineers?

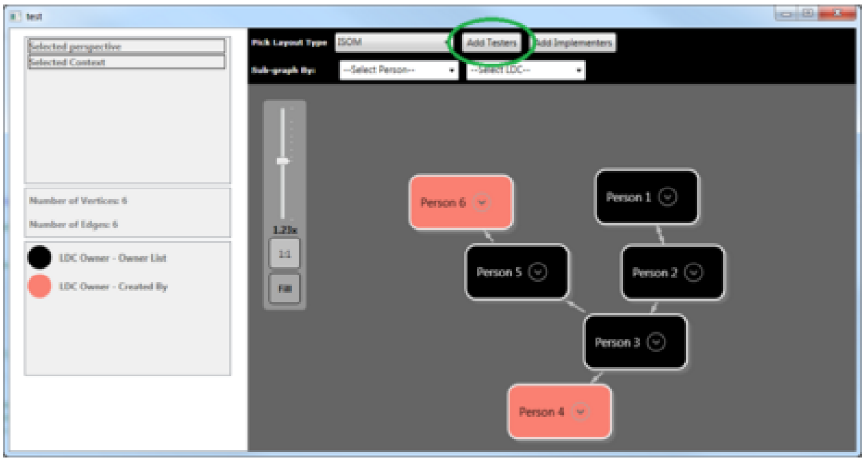
	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

What are the limitations of this feature and how can it be improved?

What possible use cases does this feature offer to your work?

Figure 8.7: Second Iteration Survey - Page 8

Adding the verification engineers to a graph -- Page 9 of 9



How accurate are the generated graphs after adding the verification engineers?

By accurate we mean how close the generated graph is to the real life social network

	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Perfectly accurate

How useful are the generated graphs after adding the verification engineers?

	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

What are the limitations of this feature and how can it be improved?

What possible use cases does this feature offer to your work?

Figure 8.8: Second Iteration Survey - Page 9

Bibliography

[1] K. Grimm, Software technology in an automotive company - major challenges, in: Software Engineering, 2003. Proceedings. 25th International Conference on, 2003, pp. 498–503.

[2] M. Aoyama, Software engineering for the connected automobiles: Opportunities and challenges, in: Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual, Vol. 2, 2015, pp. 1–1.

[3] J. Dannenberg, C. Kleinhans, The coming age of collaboration in the automotive industry, Mercer Manage J (2004) 18:88–40.

[4] P. Braun, M. Broy, F. Houdek, M. Kirchmayr, M. Müller, B. Penzenstadler, K. Pohl, T. Weyer, Guiding requirements engineering for software-intensive embedded systems in the automotive industry, Computer Science - Research and Development 29 (1) (2014) 21–43.

[5] C. Davey, J. Friedman, Software systems engineering with model-based design, in: Software Engineering for Automotive Systems, 2007. ICSE Workshops SEAS '07. Fourth International Workshop on, 2007, pp. 7–7.

[6] J. D. Procaccino, J. M. Verner, S. P. Overmyer, M. E. Darter, Case study: factors for early prediction of software development success, Information and Software Technology 44 (1) (2002) 53–62.

[7] M. Weber, J. Weisbrod, Requirements engineering in automotive development: experiences and challenges, IEEE Software 20 (1) (2003) 16–24.

[8] L. Almefelt, F. Berglund, P. Nilsson, J. Malmqvist, Requirements management in practice: findings from an empirical study in the automotive industry, Research in Engineering Design 17 (3) (2006) 113–134.

[9] S. Buhne, G. Halmans, K. Pohl, M. Weber, H. Kleinwechter, T. Wierczoch, Defining requirements at different levels of abstraction, in: Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International, 2004, pp. 346–347.

[10] B. Curtis, H. Krasner, N. Iscoe, A field study of the software design process for large systems, Commun. ACM 31 (11) (1988) 1268–1287.

[11] B. Paech, J. Dörr, M. Koehler, Improving requirements engineering communication in multiproject environments, IEEE Software 22 (1) (2005) 40–47.

[12] M. Soltani, E. Knauss, Cross-organizational challenges of requirements engineering in the autosar ecosystem: An exploratory case study, Proceedings of 5th IEEE International Workshop on Empirical Requirements Engineering.

[13] E. Otte, R. Rousseau, Social network analysis: a powerful strategy, also for the information sciences, Journal of Information Science 28 (2002) 441–453.

[14] R. Cross, S. Borgatti, A. Parker, Making invisible work visible: Using social network analysis to support strategic collaboration, California Management Review 44 (2) (2002) 25–46.

[15] T. Wolf, A. Schröter, D. Damian, L. Panjer, T. H. Nguyen, Mining task-based social networks to explore collaboration in software teams, IEEE Software 26 (1) (2009) 58–66.

[16] I. Kwan, A. Schröter, D. Damian, Does socio-technical congruence have an effect on software build success? a study of coordination in a software project, IEEE Transactions on Software Engineering 37 (3) (2011) 307–324.

[17] Volvo Group, Volvo group, http://www.volvogroup.com, last accessed Apr. 2016 (2016).

[18] SYSTEMITE AB, Systemweaver [computer program], http://www.systemweaver.se/, last accessed Apr. 2016 (2014).

[19] G. Liebel, M. Tichy, E.Knauss, O. Ljungkrantz, G. Stieglbauer, Organisational and communicational challenges in automotive requirements engineering, to be submitted to Requirements Engineering Journal.

[20] U. Abelein, B. Paech, A proposal for enhancing user-developer communication in large it projects., Proceedings of the 5th International Workshop on Cooperative and Human Aspects of Software Engineering.

[21] E. Bjarnason, K. Wnuk, B. Regnell, Requirements are slipping through the gaps - a case study on causes and effects of communication gaps in large-scale software development, IEEE 19th International RE Conference (2011) 37–46.

[22] B. Paech, K. Kohler, Perspectives on Software Requirements, Springer US, Boston, MA, 2004, Ch. Task-Driven Requirements in Object-Oriented Development, pp. 45–67.

[23] C. Y. Lin, L. Wu, Z. Wen, H. Tong, V. Griffiths-Fisher, L. Shi, D. Lubensky, Social network analysis in enterprise, Proc. IEEE 100 (9) (2012) 2759–2776.

[24] D. Damian, I. Kwan, S. Marczak, Requirements-driven collaboration: Leveraging the invisible relationships between requirements and people, Collaborative Software Eng.,J.Whitehead, I. Mistrik, J. Grundy, and A. van der Hoek, eds. (2010) 57–76.

[25] SYSTEMITE AB, Systemite ab, http://www.systemite.se/, last accessed Apr. 2016 (2015).

[26] SYSTEMITE AB, swexplorer reference manual (2015).

[27] V. Vaishnavi, B. Kuechler, Design research in information systems., Association for Information Systems.

[28] A. R. Hevner, S. T. March, J. Park, S. Ram, Design science in information systems research, MIS Q. 28 (1) (2004) 75–105.

[29] S. T. March, V. C. Storey, Design science in the information systems discipline: An introduction to the special issue on design science research, MIS Quarterly 32 (4) (2008) 725–730.

[30] S. Gregor, A. R. Hevner, Positioning and presenting design science research for maximum impact, MIS Quarterly 7 (2).

[31] P. Runeson, M. Höst, A. Rainer, B. Regnell, Case Study Research in Software Engineering, Wiley Blackwell, 2012.

[32] J. D. Herbsleb, Global software engineering: The future of socio-technical coordination, in: Future of Software Engineering, 2007. FOSE '07, 2007, pp. 188–198.

[33] J. D. Herbsleb, R. E. Grinter, Architectures, coordination, and distance: Conway's law and beyond, IEEE Software 16 (5) (1999) 63–70.

[34] K. Ehrlich, K. Chang, Leveraging expertise in global software teams: Going outside boundaries, in: Global Software Engineering, 2006. ICGSE '06. International Conference on, 2006, pp. 149–158.