LoCo CoCo: Automatically Constructing Coordination and Communication Networks from Model-Based Systems Engineering Data

Mazen Mohamad[a], Grischa Liebel[a,b,*], Eric Knauss[a,b]

[a] *Chalmers University of Technology, Gothenburg, Sweden*
[b] *University of Gothenburg, Gothenburg, Sweden*

Abstract

Context:. Communication and coordination are essential ingredients to successful requirements and software engineering. However, especially in large organisations, it is difficult to establish and maintain communication channels.

Objective:. In order to facilitate communication, we investigate automatic construction of social network models from existing requirements and systems engineering models.

Method:. We conducted a design science research study in three iterative cycles at a large automotive company, and evaluated the outcome based on 15 interviews with practitioners and a survey with 12 participants.

Results:. The resulting approach, denoted LoCo CoCo, automatically creates and visualises social networks based on selected systems engineering components of real-life, productive systems engineering models. Our results indicate that automatic construction and visualisation of social network models could be feasible and useful to overcome existing communication challenges.

Conclusion:. Despite a lack of quality in existing social data at the case company, practitioners found LoCo CoCo potentially helpful to overcome existing communication challenges. Additionally, the visualisation could trigger practitioners to keep their social data up to date.

Keywords: Systems Engineering, Communication, Coordination, Requirements Clarification, Empirical Software Engineering

*Corresponding author
Email addresses: mazenmhd@gmail.com (Mazen Mohamad), grischa@chalmers.se (Grischa Liebel), eric.knauss@cse.gu.se (Eric Knauss)

1. Introduction

Communication and coordination are essential activities in software and systems engineering [1, 2]. Breakdowns in these activities, often caused by geographical [3] or socio-cultural distances [4], can cause profound problems [5, 3]. Therefore, lack of communication and of knowledge sharing are among the top challenges in global software development [6]. In particular, communicating requirements [7], and sharing the product and contextual knowledge required to understand requirements during analysis and development [8] are challenging. At the same time, they are highly important due to the substantial impact requirements engineering (RE) has on the project outcome [9].

The structure of all interacting individuals and groups in an organisation and across organisations can be described as a social network. That is, a social structure of individuals who are related directly or indirectly to each other based on a common relation of interest [10]. Analysing such networks is known as social network analysis (SNA) and has successfully been used as a technique to uncover gaps in communication [11, 12]. In software engineering, SNA has been used, e.g., by Damian et al. [13, 14] to facilitate collaboration and relationships among individuals in software teams.

While reported results (e.g., [13, 14]) indicate that the use of SNA in software engineering is beneficial, it requires a large manual effort to create social networks. Existing automated approaches, such as Codebook [15] or ExperienceBrowser [16], focus on software development artefacts on a low level of abstraction, such as source code or bug requests. These approaches often assume that there is a large amount of data available for data mining or that there is already an ongoing discussion around artefacts such as bug requests. However, in large systems engineering projects, communication between different disciplines is required [8], taking place on a domain level independent of the source code. On this high level of abstraction, there are less available data contained in different systems engineering tools. Furthermore, there are fewer changes on and discussions around these kind of data due to long project lead times. To establish communication between disconnected parts of large organisations, our goal is to automatically construct social networks from existing data on a high level of abstraction and independent of source code, following Herbsleb's suggestion to use the *project memory* [7].

In this paper, we present LoCo CoCo, the Low-Cost Communication and Coordination approach. LoCo CoCo is the result of a one-year design science research project in three cycles, conducted at a large automotive original equipment manufacturer (OEM). LoCo CoCo automatically creates and visualises social networks from model-based systems engineering data[2] by leveraging a

[2]With model-based systems engineering data, we refer to data or artefacts related to systems engineering activities and tasks, structured by a meta model defining types and relationships between different items.

structural meta model similar to standards like EAST-ADL [17] and AUTOSAR [18]. We identify people and relations between them by extracting ownership and trace information from systems engineering data. The resulting networks are used as a supporting tool for enabling or improving communication and coordination. We evaluated LoCo CoCo analytically, by constructing social networks from real-life systems engineering data at the industrial partner. Additionally, we collected empirical data from 15 interviews and 12 surveys with practitioners.

Our results indicate that LoCo CoCo could help to address existing communication challenges by identifying important contacts across the organisation structure, thus facilitating communication of requirements and related knowledge in systems engineering. While we observed that the quality of social data in existing systems engineering tools, such as ownership data or information about who changed elements, is sometimes low, practitioners rated it as sufficient. Furthermore, visualising erroneous connections due to outdated social data could serve as a trigger for practitioners to update the data. Finally, we elicited several ethical implications arising from the use of social data. These will have to be considered when using LoCo CoCo or similar approaches in industry.

The related work regarding communication challenges in software engineering and RE, and SNA as well as construction of social networks is presented in Section 2. The research method, including a discussion of the three iterative design science cycles and a discussion of validity threats, is described in Section 3. In Section 4, the specifics of the industrial case company at which the research project was conducted and evaluated is described. We describe the three design science cycles in Sections 5 to 7. The paper is concluded with a discussion in Section 8 and a conclusion in Section 9.

2. Related Work

Communication and coordination play an essential role in software engineering [1, 2]. Due to their importance, the topics have been studied in detail since the late 80s. Based on a field study of 17 projects, Curtis, Krasner and Iscoe [5] report that communication and coordination breakdowns are a high-level problem in software design for large systems. From a survey with 775 Microsoft software engineers, Begel et al. [19] report that personal contact improved interaction between teams. In a recent systematic literature on challenges in global software development [6], lack of communication and lack of knowledge sharing among teams are among the top challenges reported. In a case study conducted in the automotive domain, we discovered several major challenges with respect to communication and knowledge sharing [8]. Among these challenges are, e.g., lack of knowledge regarding the product, lack of knowledge regarding the requirements context and establishing communication channels within the organisation (and especially across organisation boundaries).

One technique to study communication and coordination is SNA. A social network is defined as a social structure of individuals who are related directly

or indirectly to each other based on a common relation of interest [10]. SNA is, by definition, a strategy for investigating social structures by using network and graph theories [20].

SNA has been used widely outside software engineering. For example, Lin et al. [12] study the challenges and solutions in mining and analysing social networks in enterprises. The authors use multiple sources of social data, including emails, instant messages, calendar meetings and file sharing. They describe multiple challenges of this approach, such as gaps caused by not collecting teleconference data and face-to-face interaction data. By constructing social networks in Fortune 500 organisations, Cross et al. [11] could identify communication gaps and improve the situation.

In the area of software engineering, Damian et al. [13] describe an approach to construct a requirement-centric social network, representing the relationships among members working on a requirement and its associated downstream artefacts. The authors demonstrate their approach through a case study, which examines requirements-driven collaboration within an industrial, globally distributed software team. In contrast to our study, the networks in [13] are constructed manually and based on the actual communication of people. In another study, Wolf et al. [14] explore collaboration in software teams by mining repositories of broken builds. The study describes the importance of the constructed network for different roles in the project, e.g., project managers, team leaders and developers. Similar to [13], the actual communication is investigated based on how people communicate around tasks, e.g., around a filed bug. Lim et al. [21, 22] propose StakeSource, an approach that uses social networks to identify stakeholders for requirements elicitation. The approach builds on recommendations given by stakeholders regarding other potential stakeholders. StakeSource is evaluated in a university software project, with stakeholders expressing strong interest in the technique. In contrast to StakeSource, our study does not identify stakeholders for requirements elicitation, nor does it require manual recommendations from stakeholders. Instead, it only uses existing requirements and other systems engineering data with the aim to support communication, e.g., to clarify existing requirements. Mockus and Herbsleb [16] present their tool Experience Browser, which allows engineers to locate people with the right expertise. Expertise is identified by locating artefacts related to a person in software change management systems. Evaluation of the tool in a real-life context shows that users at newly established company sites are the most active users of the tool. The tool has a slightly different focus than LoCo CoCo, focusing on changes in the source code to understand who is most active in what parts of a project. As such, its focus is mainly on software developers. Begel, Phang and Zimmermann [15] present their approach Codebook, which mines software repositories to create graph representations of people and artefacts. Initially, the authors collect important use cases from a survey among developers. The approach focuses on collecting a large amount of data and inferring relationships among people through different techniques, e.g., natural language processing. To find possible connections between people, engineers can enter regular expressions describing the nature of connections they are in-

terested in. The Microsoft Visual Studio plugin CARES, based on Codebook, is presented and evaluated in [23, 24]. While the architecture of Codebook is rather general and could, therefore, also apply in our case, there are multiple differences between the approaches. First, LoCo CoCo focuses on artefacts on a high level of abstraction, i.e., independent of source code and defined items and relationships only, without mining free natural language text. This means that our approach will produce smaller networks based on actual connections in existing data, without the possibility of false positives. Hence, while Codebook collects as much data as possible and then filters these data using regular expressions, we proceed the other way around, by first selecting which connections are relevant and then visualising the entire network for exploration. Secondly, we describe in more detail the meta model underlying our graph representation and investigate additionally ethical implications of using the social networks in practice. Finally, the aim of LoCo CoCo is to establish communication between engineers at all phases of the systems engineering life cycle, while the CARES plugin focuses on developer-to-developer communication only. Especially engineers from other disciplines than software engineering are typically not aware of the source code and related implementation details. Therefore, they cannot be expected to participate in discussions around bugs or source code changes. Instead, those multi-disciplinary discussions take place at a higher level of abstraction, concerning, e.g., requirements, change requests to those requirements or system-level tests.

3. Research Method

The aim of the presented study is to aid engineers in tackling communication challenges, by automatically constructing social networks from existing systems engineering data.

This aim is broken down into the following four research questions.

RQ1: To what extent can social networks be constructed automatically from model-based systems engineering data?

RQ2: How do practitioners evaluate the potential of these networks to tackle known communication challenges in systems engineering?

RQ3: For which additional use cases would practitioners like to use LoCo CoCo?

RQ4: To what extent can LoCo CoCo be used with different tools and organisational contexts?

RQ1 aims at investigating the feasibility of LoCo CoCo by designing an artefact that automatically constructs social networks from existing data at the case company. This artefact is then evaluated with practitioners, aiming at answering RQ2 in the form of a static validation [25]. While we focus on tackling communication challenges in software engineering, further use cases might appear in different organisation contexts. Therefore, we investigate with RQ3

which additional use cases practitioners consider for LoCo CoCo. Finally, RQ4 aims at extending the scope from the case company with only one systems engineering tool to a wider audience, considering the use of multiple tools, ethical implications and visualisation.

3.1. Design Science Research Cycles

To answer the research questions, we chose a design science research method. Design science research is a problem-solving paradigm, aimed at extending existing boundaries by creating and applying new artefacts [26]. Related research methods that could have been considered for our research questions are case studies and action research. However, in case studies the focus is on observation of a phenomenon in its context, without actively designing or creating an artefact [27]. Action research has the purpose to "influence or change some aspect" [28]. As such, compared to design science, there is a stronger focus on the existing process or context at the company that is to be changed, and not on the design of a new artefact.

Following the design science guidelines proposed by Hevner et al. [26] and the cycle model described in Vaishnavi and Kuechler [29], we conducted three research cycles. Each of the three cycles was conducted using the five-step process proposed by Vaishnavi and Kuechler [29], which consists of the *awareness of the problem*, *suggestion*, *development*, *evaluation* and *conclusion* steps.

The three-cycle process is depicted in Figure 1. The first cycle, *inception*, aimed at investigating the approach by constructing and evaluating an initial artefact to automatically create social networks (RQ1 to RQ3). Based on the experience gathered in the first cycle, the second cycle, *improvement*, aimed at improving the artefact by increasing the accuracy of the constructed networks, widening its scope and by including more features (RQ1 to RQ3). The final cycle, *generalisation*, aimed at widening the scope of the constructed artefact, in order to allow technology transfer to industry (RQ4). This included proposing a concept for improved usability, an integration meta model to allow easy integration of additional data sources and studying ethical issues related to the construction and usage of social networks.

We used surveys to evaluate the outcome of each design science cycle. Surveys are appropriate to capture the current status or the current situation, and can be of quantitative or qualitative nature [30]. However, to reflect the different focus of each cycle, we collected data in different ways. We used interviews with open questions in all three cycles to allow for unanticipated answers [31] and follow-up questions. In the second cycle, we additionally used a questionnaire with mainly closed questions to allow for a quicker data collection and analysis, and to reach a wider sample. The questions asked during each evaluation cycle are listed in Appendix A. Finally, we performed an analytical evaluation of our design in the third cycle, i.e., we examined the structure of the artefact for static qualities [26] (correctness in our case).

In the first cycle, we gathered interview data through 4 semi-structured interviews with two function developers and two function owners at the case company. For each interviewee, we created 2 networks (based on two different

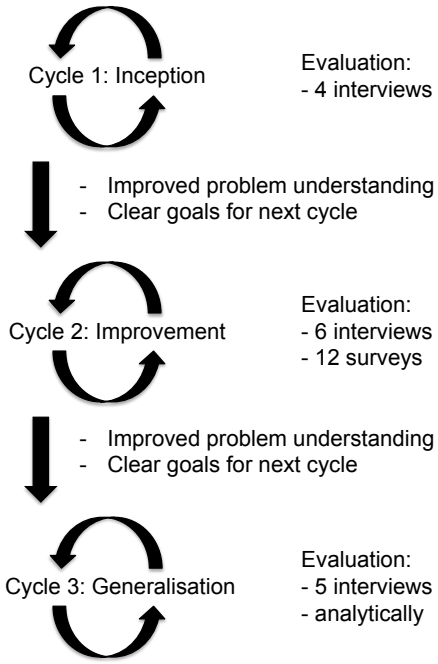

Figure 1: Three-cycle Design Science Research

sources of social data) for the interviewee's main area of work and one sub-network only depicting people with direct connections to the interviewee. We asked the participants to assess the accuracy of the networks compared to reality, i.e., whether they reflect the real-life situation at the case company. However, we use congruence instead of accuracy in the remainder of the paper, as the participants compared the network to their subjective image of reality, and not an objective reality. Additionally, we asked the interviewees to assess the usefulness of the networks, independent of their perception of congruence. We did this by first asking for the usefulness in general, without giving a concrete use case, and then asking for the usefulness for the specific use case of finding experts.

In the second cycle, we again gathered interview data from 6 practitioners, using an unstructured interview guide, loosely following the interview questions from the first cycle. The focus was on exploration of LoCo CoCo's features and, especially, on potential improvements. We complemented this with an online questionnaire, collecting quantitative data from 12 practitioners. In the online questionnaire, we briefly described the approach, followed by a number of questions on the usefulness and accuracy. We also invited the interviewees from the first cycle to participate in the questionnaire. However, as it was completely anonymous, we do not know if they participated. Additionally, to complement the industrial evaluation with an academic view, we presented our results up to

this point to a researcher not involved in the study.

Finally, the third cycle was evaluated empirically, using 5 interviews, and analytically, comparing automatically-constructed networks with manually-constructed ones. In total, we conducted 15 interviews with 11 different practitioners. Four of them were also included in a focus group, which we consulted during each cycle. Table 1 summarises the interviewees with their roles and the cycle in which they were interviewed. Here, *FG* denotes that the person participated in the focus group, whereas *C1*, *C2*, *C3* denote in which respective cycle they were interviewed. Survey participants are not listed as they were anonymous. The roles included in the evaluation steps were chosen to cover technical experts (i.e., tool experts and managers), managers with an overview of the process and engineers who would use LoCo CoCo in their daily routine (e.g., function developers or verification engineers). These different roles lead to a different work focus, providing us with different insights. We did however not favour any role over the other, e.g., by preferring people with more experience. The different amount of participants in the three evaluation cycles can be explained by a change in focus during each evaluation. In the first cycle, we aimed for early feedback with a few selected participants, mainly based on convenience. In the second cycle, we instead aimed for variety of roles, so that we would get a more general picture of the usefulness of LoCo CoCo. Finally, in the third cycle, we focused on participants with expert knowledge with respect to tools or the organisation/team structure, as the additions in this cycle were mainly of conceptual nature.

Participant	*Role*	*FG*	*C1*	*C2*	*C3*
Person A	Function Developer		X		
Person B	Function Developer		X		
Person C	Function Owner		X		
Person D	Function Owner		X		
Person E	Tool Manager	X			X
Person F	Tool Expert	X		X	X
Person G	Team Manager	X		X	X
Person H	Tool Expert	X		X	X
Person I	Function Owner			X	X
Person J	Function Owner			X	
Person K	Verification Engineer			X	

Table 1: Interview participants with role

A detailed discussion of what we did in each cycle and of the results is presented in Sections 5 to 9.

3.2. Validity Threats

In the following, we discuss existing threats to validity and our mitigation strategies.

3.2.1. Construct Validity

In order to avoid misunderstandings, we presented the purpose of the study prior to each interview and in the survey invitation email. In the online questionnaire, we provided a screen shot for each set of questions to clarify which sub-feature of LoCo CoCo we were evaluating. In the third cycle, all interviewees had previous knowledge of the project. We presented a brief introduction at the beginning of each interview to remind them and to avoid confusion.

The evaluation presented in this study is a static validation according to the classification by Gorschek et al. [25]. Therefore, we cannot state that LoCo CoCo does indeed aid in tackling communication challenges in real-life industrial practice. We plan to conduct this kind of validation in an ongoing project.

3.2.2. Internal Validity

LoCo CoCo produces potentially sensitive output, e.g., key persons or isolated persons in a given context. Even though the networks reflect relationships existing in real-life data, it is possible that the practitioners' feedback is affected by this sensitivity. This sensitivity is a potential threat to validity and cannot be ruled out in our current approach.

3.2.3. External Validity

The study was conducted in one automotive company, constructing networks from models in one systems engineering tool. Even though we devoted the third design science cycle to increase the generalisability of the approach, it might so far be limited.

To improve external validity, we chose interviewees and survey participants from different departments and roles at the case company.

3.2.4. Reliability

Our tool choice and access to the case company might be a decisive factor for the success of our study. A lack of similar conditions could potentially limit reliability.

To enable replication, the questionnaire and the survey raw data are furthermore published[3].

4. Case Company

Next, we introduce the case company and the systems engineering tool SystemWeaver[4], which we used as a systems engineering repository during the first two cycles.

[3] `http://grischaliebel.de/data/research/MLK_LocoCoco.zip`

[4] http://www.systemweaver.se/

4.1. Volvo GTT

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 [32]. Volvo Group Truck Technology (GTT) is the research and development organisation of Volvo Group, with over 7000 employees working in global teams. This study was conducted in collaboration with the Electrical & Electronics Engineering (E&EE) department at Volvo GTT. The department uses the tool SystemWeaver as a systems engineering environment. Engineers in the department receive abstract, implementation-independent requirements for each project from outside the department. These requirements are then broken down within the department into smaller, detailed parts and, ultimately, into logical components. 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 organisation starts to prepare the verification activities independently of the 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, and to establish tracing between these artefacts.

4.2. SystemWeaver and its Architecture

SystemWeaver is an information management solution for systems engineering and software development [33], developed by Systemite AB[5]. SystemWeaver allows users to build a single model, which encapsulates the system description and provides different views on that model. Users cannot build anything that is not explicitly allowed by the underlying meta model. In order to fit different domains or company profiles, this meta model can be adapted.

The top part of Figure 2 represents the relevant part of the meta model of SystemWeaver's conceptual architecture used in this study. The meta model consists of the concepts *item, part, object* and *attribute*. The *item* is the smallest reusable object in SystemWeaver, e.g., a requirement or a component. The *part* defines a connection between two *items*, practically a trace link. The *attribute* is a typed value for an object. This value is unique for the object and cannot be used or shared by other objects. Examples of attributes include identifiers, names or descriptions.

The bottom part of Figure 2 outlines how testing and verification data are stored, and how test and design architecture interact.

4.3. SystemWeaver at Volvo GTT

Volvo GTT has used SystemWeaver since 2007 as a platform for systems engineering activities. Volvo GTT categorises data in SystemWeaver based on

[5] http://www.systemite.se/

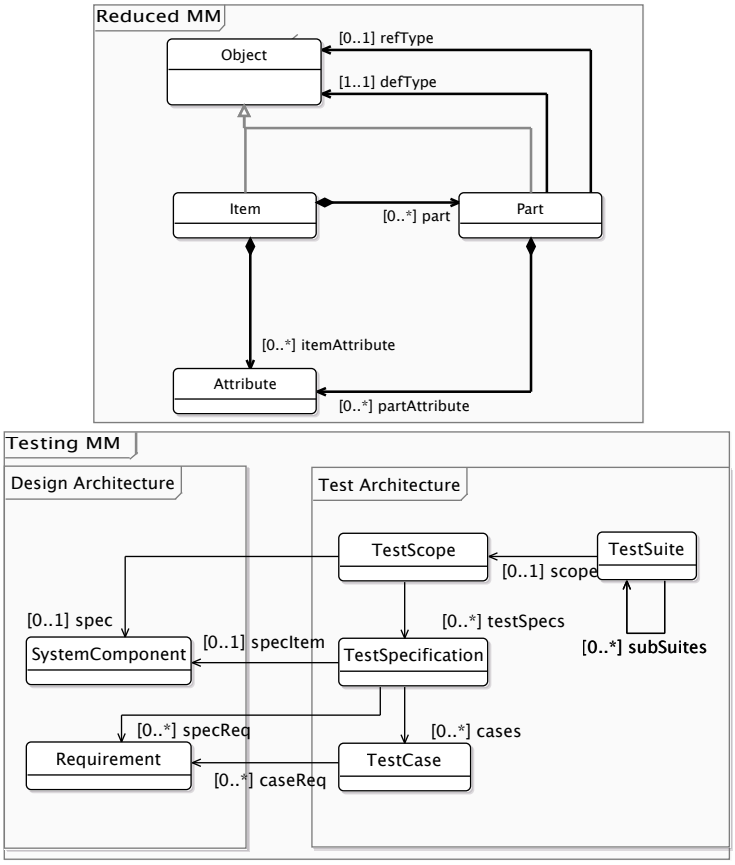

Figure 2: Meta models of the conceptual architecture [34] and testing and verification

multiple architectural viewpoints, based on the EAST-ADL meta model [17]. These viewpoints are explained in the following sub-sections.

4.3.1. End-to-End Functions

An End-to-End (E2E) function is an artefact that specifies and represents an end-user task, which transforms a user input into an output. An E2E function includes requirements for this task. Each E2E function has a clear purpose, a defined scope and involves at least one human actor.

4.3.2. Truck Application System

The Truck Application System represents the abstract design structure of the system. It contains Logical Design Architecture (LDA) elements. The functionality of LDAs is defined by so-called Logical Design Components (LDC). An LDA contains the connections between the LDCs, as well as input and output

ports for signals exchanged with other LDAs. In turn, an LDC may contain items, e.g., local requirements or send/receive ports. For the sake of simplicity, we call LDCs *logical components* in the remainder of this paper.

4.3.3. Installation System - Executable Software System

The installation system represents how each of the logical components is allocated to hardware. This design consists of Real Allocation Targets (RATs), which link 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 RATs.

4.3.4. Collaborations

Collaborations are sets of logical components that are connected to each other and interact in a specific way. The purpose of a collaboration is to provide the connection between an E2E Function and the design in the end product. The collaboration defines how logical components interact to realise an E2E Function. 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 testing organisation uses the requirements found in the collaboration to create the test cases.

5. Cycle I - Inception

5.1. Awareness of the problem

The awareness of the problem in the first cycle is directly derived from the motivation and related work for this paper. Communicating requirements is challenging [7], but highly important [9]. In particular, our own study revealed several key communication problems related to the understanding of existing requirements [8].

In discussions with the focus group of experts at the case company, these problems were confirmed. As it was given special emphasis by the focus group, we decided to specifically focus on the use case "UC1: An engineer is trying to find experts for clarification of requirements or design".

5.2. Suggestion

In order to address the problem described above, we decided to design our artefact in the form of a support tool, LoCo CoCo. This tool aims to visualise expert networks, reflecting the structure of existing systems engineering data at the company.

To construct these networks automatically, we decided to extract data from systems engineering tools at the case company. For the first cycle, we chose SystemWeaver as a systems engineering tool, as it contains data covering a large proportion of systems engineering activities at the case company. Thus, we cover a large part of the roles in the studied department and reduce accidental complexity introduced by multiple tools.

While networks constructed by LoCo CoCo could be used for multiple purposes, we decided to focus explicitly on UC1. In the evaluation step, we then additionally explored the potential value of LoCo CoCo for additional use cases.

5.3. Development

To construct the first version of LoCo CoCo, we had to make three key development decisions: the method of data extraction (how to obtain the data), the network visualisation (how/where to draw the network) and the network construction (how to construct a social network from the existing data).

SystemWeaver supports several different data extraction methods, i.e., an XML-based reporting language, XML export and a C# API. At the same time, SystemWeaver allows embedded visualisation of networks through an XML-based reporting language. Therefore, using the XML-based reporting language and the embedded visualisation allowed us to embed LoCo CoCo directly in SystemWeaver, without the need for a third-party tool. As this would simplify deployment and evaluation, and at the same time raise the acceptance at the case company, we decided to use this combination of techniques for data extraction and network visualisation.

For network construction, we identified two sources of social data in SystemWeaver, a property in each item representing the owner of the item (*owned by*) and another item property representing the last user that committed a change to the item (*last changed by*). These attributes allowed us to connect systems engineering items, such as requirements or software components, to people.

However, the large amount of data in SystemWeaver at the case company prohibited to simply use all existing items for the construction of social networks. Therefore, we had to find the right abstraction level to avoid networks with a large amount of nodes and incomprehensible relationships, or trivial networks containing only single nodes. The focus group helped us to identify logical components (see Section 4.3.2) as the central working part of the daily engineering work. Therefore, we chose these as the 'right' abstraction level. That is, we used people who owned or last changed logical components as the nodes in our social networks.

To connect our nodes in the social network, we used connections between logical components. These connections exist in two different ways at the case company, implicitly and explicitly. Two logical components are implicitly connected if they share the same signal, one component having it as an input signal and the other one as an output signal. Explicit connections are expressed using a trace link (a connector item with two parts in SystemWeaver). Figure 3 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 are implicitly connected by sharing the signal M (input to one and output to the other).

To create a single network, we did not extract all logical components in the entire case company database, but only logical components contained in

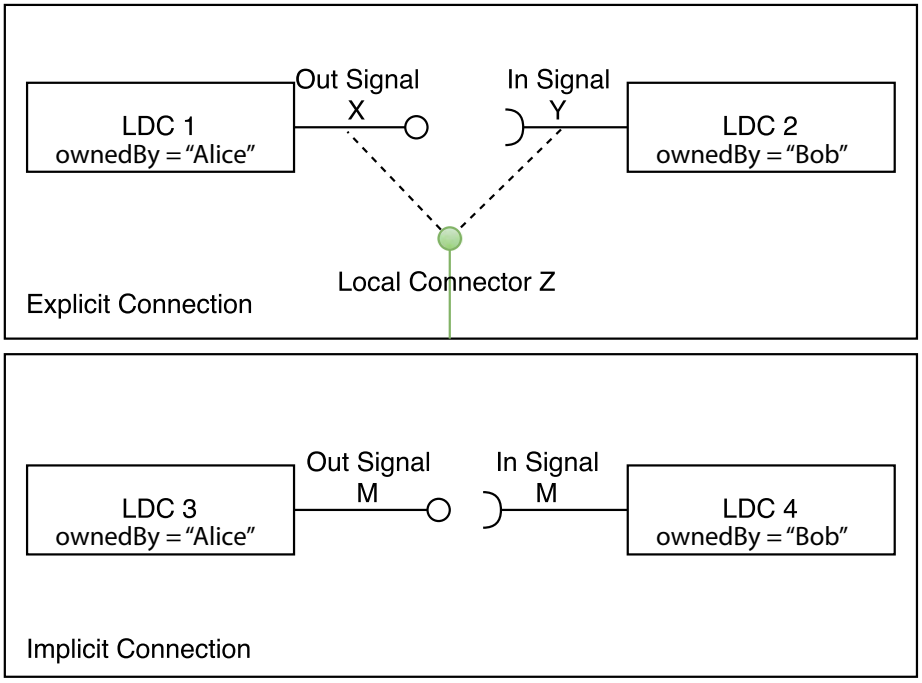

Figure 3: Example of implicit and explicit connections in SystemWeaver

selected viewpoints, e.g., the logical architecture. The production database of the case company contains over 300 of such viewpoints. In turn, these contain approximately 150 logical components, 300 functional requirements and over 4000 test cases. Traversing this amount of data and constructing the network proved to be feasible and took less than one minute in all cases, even for the largest viewpoint in the database. As an organisation structure is typically a reflection of the product it is building (Conway's Law [3]), we reasoned that limiting the scope in this way would produce networks that are limited to one organisation unit. Therefore, in contrast to a larger scope spanning multiple organisation units, we expected that practitioners would be able to assess the congruence of the resulting networks.

We created two networks for each main component, one network using the *owned by* property and one network using the *last changed by* property. To create edges between the network nodes, we used both implicit and explicit connections. Duplicate edges and the direction of the edges were omitted in the resulting network. This was a design decision to simplify the user interface for evaluation.

As an abstract example of how the extraction works, consider the four logical components LDC1 to LDC4 in Figure 3. Starting from the first logical

component, LDC1, we would add the owner of that component, Alice, to the social network. Then, we would iterate over all connectors of LDC1, returning a connection to LDC2. Therefore, we would add the owner of LDC2, Bob, to the social network and connect Alice to Bob. Continuing the iteration with LDC2 would not lead to any changes in the network. For LDC3, we would not add any additional person to the social network, as Alice is already present. However, we would add one more connection between Alice and Bob since LDC3 has a connector to LDC4.

To illustrate the amount of data used in our approach, we selected one so-called allocation component to construct a social network. This component yielded a social network containing 33 people (i.e., nodes) and 100 edges (13 of which were duplicate), and was constructed in less than a minute. Among the people, we discovered 2 completely isolated people, two people that were only connected to one another, i.e., that formed a sub graph of size 2, and two people that only had a single connection to another person.

To assess the direct connections of a practitioner and limit complexity, we created a script to extract sub-networks for a single person and all his/her direct contacts.

5.4. Evaluation

For each interviewee, we created two networks each (based on the two different sources of social data) for the interviewee's main area of work, and one sub-network depicting people with direct connections to the interviewee. The interviewees gave an average rating of 4 out of 10 for the congruence of the networks based on the *created by* property and a 4.5 out of 10 rating 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 improve the congruence of the networks, interviewees proposed a number of different data sources. These ranged from data in different tools, such as the department's issue tracking tool, to different properties within SystemWeaver. In particular, a manually created list of component ownership within SystemWeaver was named by the interviewees. This list was adopted as a data source for the second cycle. This discussion indicates that social data is secondary in systems engineering tools, as there is initially no clear need for having the data. However, when a need to use these parts of the data arises, workarounds such as the manual owner list are created. Thus, establishing clear use cases could encourage to keep social data up to date.

The usefulness of the resulting networks was rated as 2 out of 10 on average. Several interviewees stated that they already knew the network we showed them and would, therefore, not need a tool. We relate this to the fact that we constructed networks for a specific role and, thus, target people who are co-located in the case company. This indicates that the value of constructing social networks for communication purposes should increase once the network

crosses organisation boundaries. Several interviewees requested more information about the network, e.g., why a certain edge in the network existed.

Interviewees were not able to suggest possible use cases for LoCo CoCo. However, when we provided them with UC1, they emphasised that LoCo CoCo could be a suitable option to identify experts, especially for new employees. Hence, junior employees might evaluate the usefulness of LoCo CoCo higher than senior staff, who have already built up a network within the organisation.

6. Cycle II - Improvement

6.1. Awareness of the problem

From the first cycle, we realised that we had to enhance the congruence of the data by using additional data sources. Furthermore, we learned that, to increase the usefulness of social networks, they have to show connections across organisation boundaries. In order for the networks to be clearer to the users, we had to make additional information available, e.g., explanations for connections between people. Finally, for evaluating networks, we needed to provide clear use cases to the interviewees.

6.2. Suggestion

Our focus in the second cycle remained on UC1.

The source of social data was one of our main concerns. Therefore, we had to improve the network congruence by using a third source of social data, the manual owner list, which was raised by the interviewees. Additionally, we aimed to extend the scope of LoCo CoCo by constructing networks that depict relations across organisation boundaries. To make LoCo CoCo more useful, we decided to add a way for users to select from which main component (container) a network should be constructed, a purely manual task in the first cycle. Finally, we added a layer of information behind the actual social network. Providing such an information layer meant that we needed an interactive GUI. This improvement would make it possible to provide additional information by extracting sub-networks based on different criteria, e.g., a specific person or component.

During the evaluation, we provided three use cases, in addition to UC1, and asked participants whether or not they would use LoCo CoCo for these use cases. Similar to UC1, we elicited these additional use cases with the focus group during the first cycle, but ultimately did not use them as the primary use case.

6.3. Development

In order to allow for an interactive approach, we had to abandon the SystemWeaver graph library and change LoCo CoCo to a standalone application. Therefore, we developed a `C#` standalone application, accessing the data in SystemWeaver using the `C#` API. The API provides freedom to transform the systems engineering model into any format, use different sources to construct the

data set and freely select the presentation format of the networks. Hence, our standalone artefact can also be used to construct networks in XML format that can be visualised directly in SystemWeaver.

For visualisation in the standalone artefact, we used the library GraphSharp [35]. Additionally to the functionality from the first cycle, we added multiple features to LoCo CoCo, which we describe in the following section[6].

6.3.1. Improved congruence and information

Each node in a LoCo CoCo network corresponds to an owner of at least one logical component. In contrast to the first cycle, ownership was determined by filtering the logical components according to their existence in the owner list. In case a logical component was not referenced in the owner list, we used the *created by* property as in the first cycle. This was the case in roughly half of all components.

In order to better understand the created networks, we provided further information, i.e., the source of the network (the container item), the number of nodes and edges, and a legend for the different node types and colours. For each node, we provided a list of all logical components owned by that person. Additionally, we calculated edge weights as the number of connected logical components owned by the vertices of an edge and provided it as a tool tip in LoCo CoCo. Figure 4 shows an example of a created network in the standalone artefact. On the left side, the tool provides information regarding the currently selected source, the number of nodes and edges of the network, as well as a legend of the different colour codes (black for owners of components, red for creators of components, yellow for implementers). Options to change the layout and to view sub-graphs are shown in the top of the tool.

6.3.2. Adding data from verification and implementation

To visualise people across the organisation structure, we decided to include data from implementation and verification. While verification and implementation items are included in SystemWeaver, traces between logical components and verification items (e.g., test cases) are unidirectional from verification items to logical components. Therefore, to add edges from owners of a logical component to owners of verification items, we had to traverse the entire verification model. As the very large data set made this computation a very time consuming task, we decided to only implement this feature for a subset of the components. In the future, we plan to improve the performance by extracting the entire model once and storing the trace links as bidirectional links.

Similar to verification, implementation items for in-house software are contained in SystemWeaver. However, at the case company, a different database is used for all implementation items ("implementation database"). This database is not connected to the database used for verification and analysis items ("analysis database"). The only connection between the two databases are links on

[6] We implemented a total of six new features, but only present three here for sake of brevity.

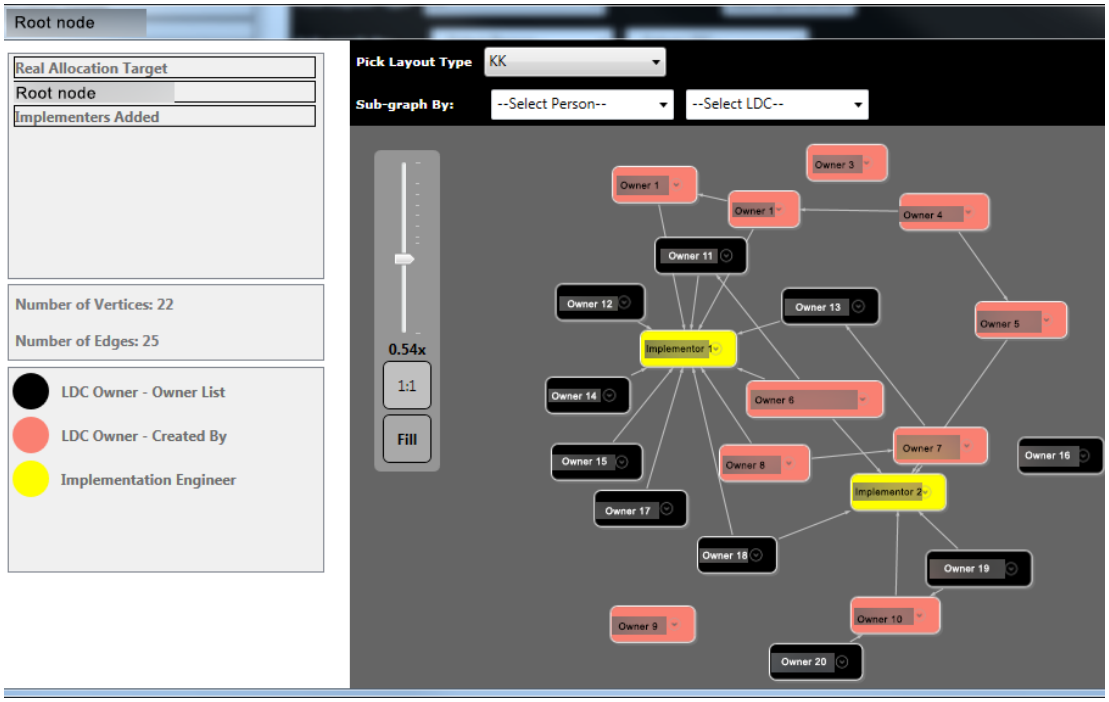

Figure 4: Example of a network created in LoCo CoCo.

the requirements level in the implementation database, carrying identifiers of the items in the analysis database. Figure 5 outlines how LoCo CoCo uses these links to find connections between people in both databases. First, LoCo CoCo constructs the network from the analysis database as in Cycle I. Then, it searches all the requirements in the implementation database which have links to the analysis database. Whenever such a link is found, the owner of the implementation unit containing the linked requirement in the implementation database is connected to the owner of the logical component containing the respective requirement in the analysis database. To lower the execution time, we also introduced a hard-coded link between the main components in both databases.

6.3.3. Creating networks based on user-selected contexts

In the first cycle, we manually chose main components to construct several social networks. To automate this process and make it adaptable to a user-specific context, we added a feature that allows users to select a main component as a root node for data extraction. The resulting network is then only constructed using logical components contained in this component.

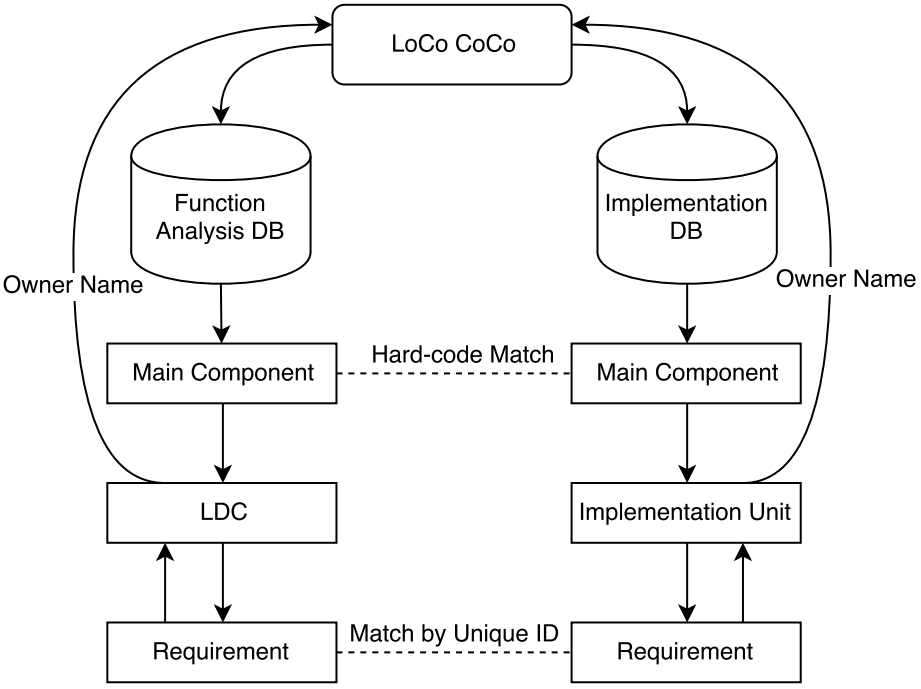

Figure 5: Adding implementation engineers in LoCo CoCo

6.4. Evaluation

We evaluated the usefulness and the congruence of LoCo CoCo in general and of the newly implemented features. Additionally, we asked practitioners about limitations of, and possible add-ons for, LoCo CoCo.

6.4.1. Usefulness of LoCo CoCo

The survey data show that LoCo CoCo is considered to be useful by most of the participants, as depicted in the box plot in Figure 6. Nine out of twelve participants find the approach useful or somewhat useful.

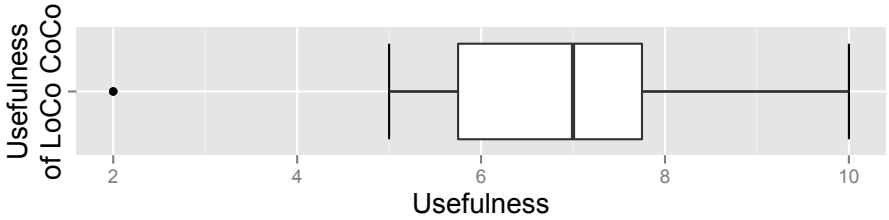

Figure 6: LoCo CoCo usefulness (0 = not useful, 10 = very useful)

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 UC1 received the highest rating (eight participants), the remaining use cases received similar ratings. Both "Finding out who will be affected by a committed change" and "Gathering the required persons for a meeting set-up" were selected by 7 participants and "Finding the persons with most context knowledge" by 6 out of 12.

With respect to the usefulness of LoCo CoCo, the interview data were aligned with the survey data. All six interviewees felt that LoCo CoCo is useful and that there are multiple use cases it can be used for. In contrast to the first cycle, several new use cases were provided. Two of the interviewees stated that LoCo CoCo provides an easy way to get instant feedback of the complexity of different components. This information is often known according to the interviewees, but visualising it from a social perspective provides additional knowledge. In addition, the interviewees felt that LoCo CoCo could provide a good way to visualise 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. These two suggestions are of different nature, as the first provides context knowledge (i.e., complexity of a component) and the second provides an overview (i.e., isolated nodes in the overall communication).

The overall usefulness assessment by practitioners in the second cycle is, therefore, considerably more positive than in the first cycle, indicating that the improved congruence did indeed raise the acceptance. Additionally, interviewees showed more interest in the networks as in the first cycle. This could either be due to the increased usefulness and congruence, or due to the fact that the tool allowed interaction.

6.4.2. Limitations of LoCo CoCo

Similar to the first cycle, congruence was still mentioned as a limitation, even though the usefulness was evaluated much higher. Seven participants stated that the ownership data were not up to date in SystemWeaver, which led to false information in the generated networks. 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 associated to the items in the first place. However, the interviewee revealed that the reason for the ownership data to be outdated, even in the owner list, is often that *"it is considered as a low priority to update these data compared to other core business tasks"*. This low priority stems from the fact that the data are currently not used for any purpose. Introducing LoCo CoCo or a similar approach that makes use of the data and provides a benefit to the engineers, we believe that they could be triggered to update the ownership data. This was also confirmed by several interviewees. Additionally, easy changes in SystemWeaver, such as logging all changes to an item and updating the last

changed field automatically, could make the networks highly congruent, without any manual effort from the user side.

According to two survey participants, another limitation of LoCo CoCo is that it is developed standalone. The participants felt that it should be integrated into SystemWeaver. While the decision to have a standalone artefact resulted from restrictions in SystemWeaver, we believe that our close collaboration with the vendor could make an integration possible in the future. However, this is clearly not the case if multiple tools were integrated as data sources into LoCo CoCo. Hence, the decision to move to a standalone application was not in all cases seen positively.

Three participants felt that they had more knowledge of whom to contact than what was offered by LoCo CoCo. Hence, they 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. Additionally, we think that experienced engineers could still benefit if more cross-organisation data were added to LoCo CoCo.

6.4.3. Evaluation of Features

Additionally to the general evaluation of LoCo CoCo, we evaluated the features we added in the second cycle. These are:

- One main feature: creating a network based on a specific content.
- Two context-broadening features: adding implementation and verification engineers.

For all features, we asked for usefulness and congruence. Figure 7 shows box plots for the usefulness and the congruence of each of the three features. The answers are on a scale from one to ten, where ten is very useful/congruent and one is not useful/congruent at all. The medians of the features are similar for both usefulness and congruence. However, no feature scored a mean rating lower than five, which we believe indicates the potential of the features' usefulness and congruence. Interestingly, all box plots but one have a bandwidth covering the entire scale. We believe that this could be related to the congruence of subnetworks: While the overall network could be highly congruent in some parts, leading to a high evaluation of congruence if the evaluator knows that area, other parts of the network might have low congruence.

As Figure 7 indicates, there is a strong correlation between the usefulness and congruence answers for all features. These are listed in Table 2.

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 congruence of a feature if they value the usefulness of it. Additionally, participants who initially observe a low congruence, e.g., by identifying missing links between themselves and other engineers, tend to rate the usefulness lower. While this correlation indicates that not all usefulness and congruence answers

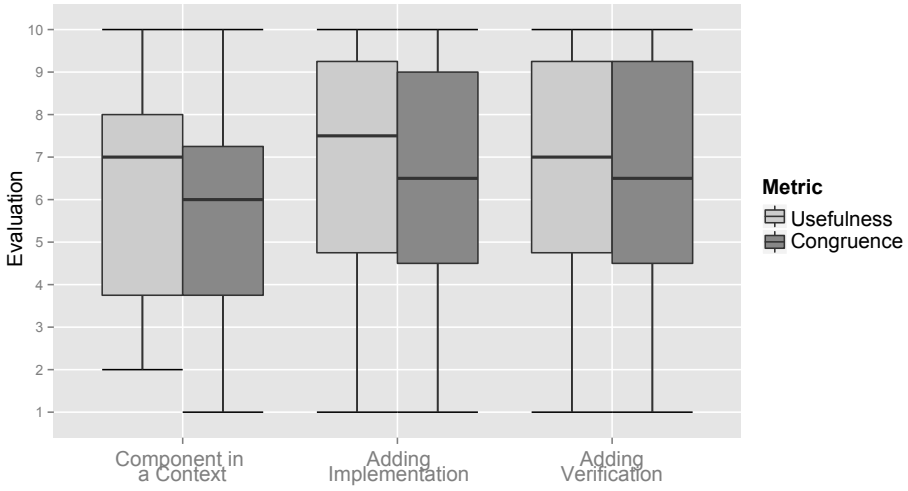

Figure 7: Average usefulness and congruence of different features

Table 2: Correlations between Usefulness and Congruence for LoCo CoCo Features

Feature	**Spearman's ρ**
Component in a Context	**0.7089***
Adding Implementation	**0.8434****
Adding Verification	**0.9214****

*p<0.01, **p<0.001

are in fact appropriate, they support the observation that the congruence is highly affecting the usefulness of LoCo CoCo.

6.4.4. Enhancement of LoCo CoCo

To make the approach more useful, survey participants suggested a number of enhancements and features to implement. 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 these data is currently not available, this enhancement would depend on a decision by the case company or tool provider. However, it is valuable feedback for using LoCo CoCo with other tools than SystemWeaver.

Four interviewees 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 who 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.

Apart from the interviews, we also demonstrated the results to a researcher. Regarding more general feedback, he raised that the visualisation would need improvement. For example, he named the use of street-light colours as a drawback, as they have an implicit semantics to many people (e.g., red being negative).

7. Cycle III - Generalisation

7.1. Awareness of the problem

The second cycle showed that the congruence and usefulness of the networks were considered sufficient, despite the fact that social data are not a primary concern. Therefore, an important question at this point in time is if LoCo CoCo can be extended to use data from systems engineering tools different from SystemWeaver. Additionally, the previous cycle raised ethical concerns and drawbacks with the existing design approach.

7.2. Suggestion

During the third cycle, we decided to focus on the generalisation of LoCo CoCo. First, we decided to incorporate data from additional systems engineering tools, thus, leaving the pure SystemWeaver environment behind. To do so, we decided to construct a common meta model for social networks extracted from different tools, considering different ways of data extraction, data persistence and the availability of social data in other tools. We would then evaluate this meta model by extracting data from one additional tool and integrate them into our social networks. Secondly, we aimed to improve the network visualisation, following established guidelines from the usability community. We focused on those parts of our artefact that were raised as problematic by a peer. These included the layout, symbols and their colour coding, and the presentation of information in different sub-graphs. Finally, we decided to further investigate ethical implications of visualising social networks, in particular when combining data from several tools. This investigation was performed during the evaluation phase only, as we decided to collect qualitative data in the form of interviews.

7.3. Development

We started the third cycle by building a common meta model for our social networks, the LoCo CoCo meta model. This model is depicted in Figure 8. Nodes represent all the items imported from different sources, e.g., logical components. DataSets are the source of the data, e.g., a SystemWeaver database. The MainNodes represent the nodes on which the social networks will be built, i.e., the containers. Attributes describe arbitrary attributes and properties of a node, e.g., identifiers or creation dates. Persons are the actual people in the network, related to Nodes in the network. DataSetConnections represent how different DataSets are related to each other, e.g., describing the hard-coded links between issues and logical components. Each connection consists of two

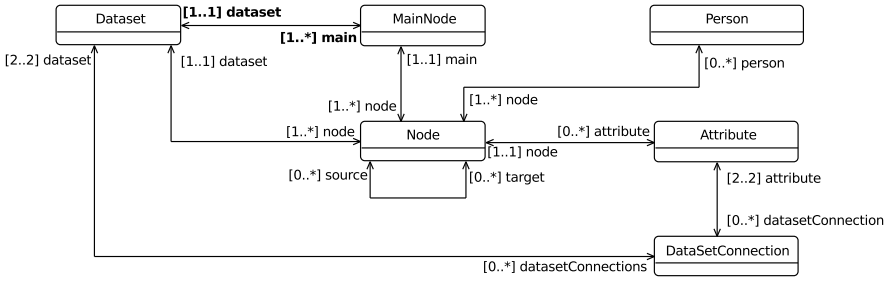

Figure 8: LoCo CoCo meta-model

DataSets and is identified via two attributes, one from each DataSet to be connected. Finally, the Node's target and source relations describe connections between nodes within one DataSet.

As the issue tracking tool IBM Clear Quest[7] was in widespread use at the case company, we decided to use for evaluating the meta model. It contains information about people who create or are assigned issues and is furthermore linked to SystemWeaver, as items with a special attribute referring to Clear Quest issues are created manually in SystemWeaver. Furthermore, Clear Quest allows file export. Therefore, it is suited for integration in LoCo CoCo. After extracting data from both SystemWeaver and Clear Quest, we transformed them into the LoCo CoCo meta model. We created two model transformations and implemented them in a `C#` application, one for each tool. In the SystemWeaver case, we transformed logical components and issues (manually created references to Clear Quest issues) into nodes of LoCo CoCo. Figure 9 exemplifies this transformation. Similarly, we transformed both the implicit and explicit connections of the logical components into LoCo CoCo node connections.

We transformed each issue in Clear Quest into a node in LoCo CoCo and the social data into persons. Issue IDs were added as attributes to the nodes. We configured the DataSetConnection based on the manual connections from SystemWeaver to Clear Quest. With that, we could feed the data into LoCo CoCo for visualisation.

To improve visualisation, we created a conceptual design for LoCo CoCo. As our focus was purely on obtaining feedback, we did not implement the design in this cycle.

To make use of the available horizontal space of modern displays, we dedicate the left side of the screen to the graph, while showing overview data on the right side. The eye movement follows a top-to-bottom and left-to-right path throughout the interface [36], at least in Western cultures. Therefore, our design

[7] http://www-03.ibm.com/software/products/en/clearquest

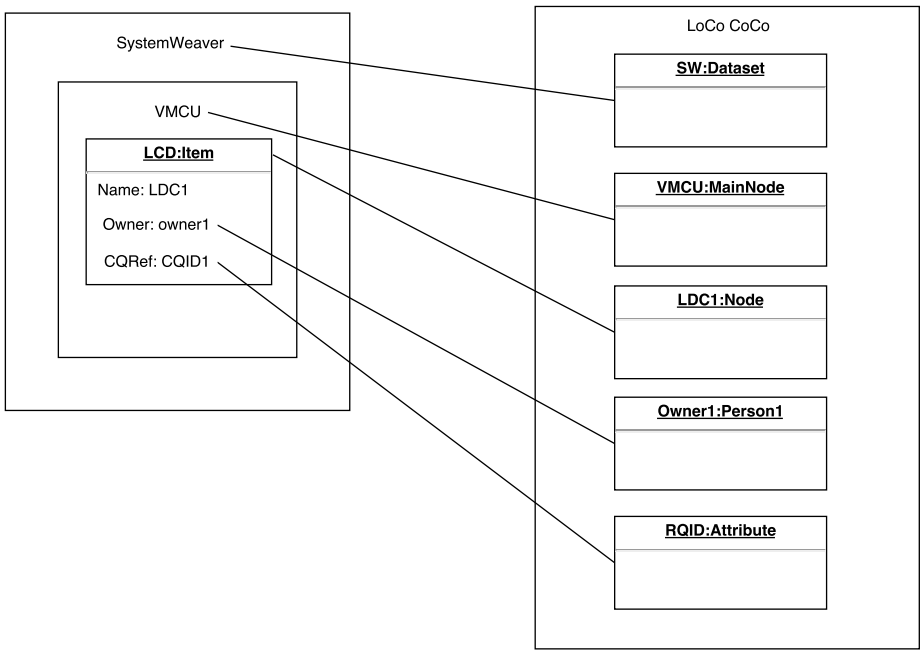

Figure 9: SystemWeaver logical component to LoCo CoCo model transformation

displays the graph on the left side, followed by the tab selectors, a search box and a list from top to bottom. Each tab contains a search box and a list view for items. In order to strengthen the mental connection [36], we use symbols to distinguish between persons and items. A simplified human silhouette represents person nodes and the puzzle piece represents item notes.

Figure 10 shows the initial visualisation view. Each person in this view is represented by a name tag, which contains a colour-coded tag to indicate the data source, followed by the person's name and a "puzzle" button to show a list of items owned by this person. We colour-code only the top three sources and group the rest in a single "Other sources" category. The colour palette was selected from low-saturation, high level (luminance) colours, as these are neutral and do not draw unwanted attention as much as highly-saturated colours [36]. The hues (blue, purple, orange and cyan) are widely separated in the colour space, making it easy to visually distinguish the data source for each person [37].

Figure 11 depicts the sub-graph of a connection between two persons. The user reaches this view by clicking on a connecting line between two nodes. Because this view contains a different type of data, i.e., items instead of people, we chose to hide the original graph and display this view on a solid background. The persons sharing these items are highlighted in the list.

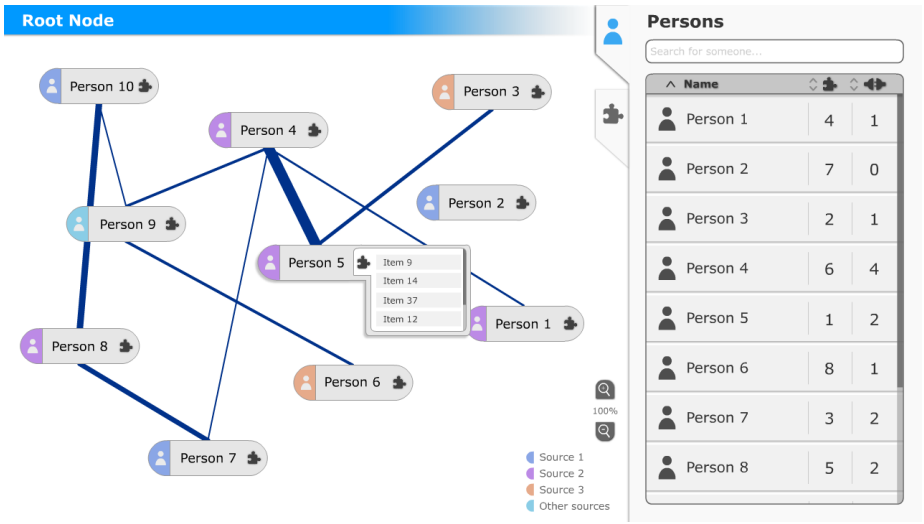

Figure 10: LoCo CoCo overview graph with a person's list of owned items

7.4. Evaluation

Our goal in the evaluation phase was to assess the feasibility and limitations of the tool integration and the LoCo CoCo meta model, compare the visualisation concept with the old design, and discuss ethical implications of LoCo CoCo.

We split the evaluation into two parts. First, we evaluated analytically our integration model by creating social networks from two data sources. That is, we created a social network based on the extracted data from SystemWeaver and Clear Quest, and compared it to a network that we created manually. The network was created by the first author of this paper, who had approximately 10 months of experience with the used tools and the case company. We were able to construct the transformation from the original tool data to LoCo CoCo models. We then manually created a social network for one of the main systems engineering components, using the same construction approach as the tool, and compared it to the automatically created network. This yielded a perfect match.

As a second evaluation part, we conducted interviews with five practitioners, all of which had previous knowledge about LoCo CoCo. Each interview lasted one hour, starting with a presentation of the visualisation concept, followed by three sets of questions. The first set of questions was related to the use of different systems engineering tools and the feasibility of transforming data from these tools into a LoCo CoCo model. Furthermore, we asked questions regarding the quality of the LoCo CoCo meta model and possible ways to improve it. We then proceeded to assess the visualisation concept, both by asking general questions and comparative questions with respect to the old design. Each interview was concluded with ethical questions, e.g., how the tool could be misused or how

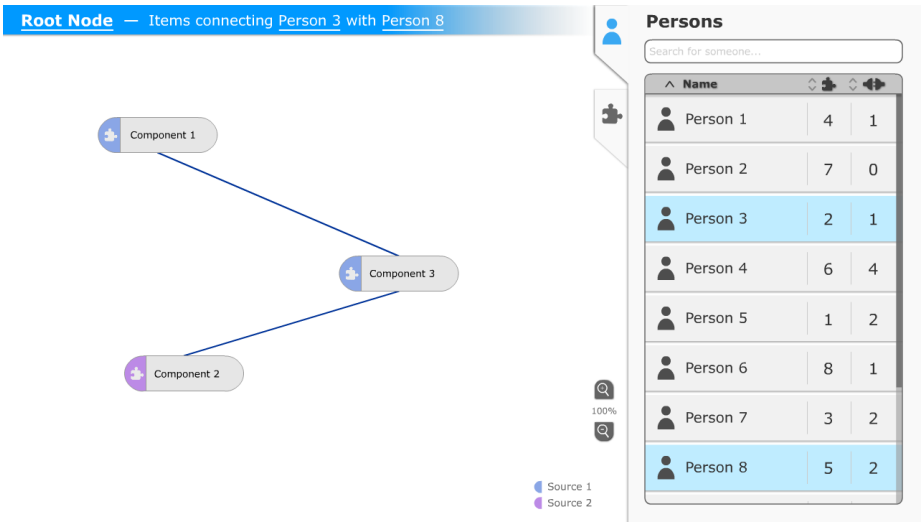

Figure 11: LoCo CoCo sub-graph based on a connection between two persons

interviewees feel when they see themselves as isolated or connected nodes in the network.

Four interviewees agreed that tools they use on a daily basis could be integrated into LoCo CoCo by using the proposed meta model. One interviewee stated that the main challenge would be that the tools rely *"heavily on the process and not paying attention to the potential and power of the persons"*. That is, person-related data, such as who changes or owns items in the tools, are not considered important. This aligns with the finding from the first two cycles, that social data are secondary and can, therefore, be outdated.

One of the interviewees mentioned that there have been projects to integrate tools, but synchronising data was always a problem. Therefore, we think that it is important that LoCo CoCo should continue to only read existing data, without modifying them.

All five interviewees preferred the proposed design to the previous one, especially the colour coding. Other feedback we obtained was that the new design looked much *"cleaner"* and *"more professional"*.

According to our observation, the ethical questions caught the interviewees' attention most. All interviewees agreed that the tool might be misused by different types of users. They proposed the following misuse cases for LoCo CoCo:

- Managers can use the application to judge the performance of their employees. Data taken directly from a systems engineering tool might suggest that one person works more than another, even if this is not the case.
- Employees may pick items that make them appear as central nodes in the

networks, as a consequence to the previous scenario. This phenomenon could be related to the concept of *"Evaluation Apprehension"* [30], i.e., that persons react according to the way they are evaluated.

- The tool might be used to blame others, e.g., to blame an owner of a component for the delay of a task.
- The tool might be used to misinterpret how important a person is, e.g., if one person is connected to a manager and another person is not.

All five interviewees had no problem of seeing themselves as a part of a social network. However, when asked specifically how they would feel if they were isolated in such a network, two of them stated that it might be discouraging. Being heavily connected was considered good for three interviewees, while the remaining two stated that they might feel nervous about it.

All five interviewees felt that there is no problem of having their information appear in LoCo CoCo *"as long as it is internal"*, especially as the information already exists in other tools.

8. Discussion

In the following, we discuss the answers to our four research questions and relate them to existing work.

Regarding RQ1, *To what extent can social networks be constructed automatically from model-based systems engineering data?*, our results indicate that this is indeed technically possible. To construct any such network, LoCo CoCo requires existing model-based systems engineering data, i.e., structured data with typed items and relationships among them. In particular, our evaluation indicates that data spanning multiple organisation units would be required to make the approach useful. This means that a tool containing test cases linked to requirements in another tool would already be sufficient. We would expect that such data exist in any project or organisation context large enough so that communication problems arise. Therefore, the amount of data available should not be a restricting factor. One exception might be teams working in an agile fashion, as they strive to reduce documentation as much as possible. However, related work on large-scale agile, e.g., [38, 39], indicates that given sufficient scale and complexity, there is still a need for documentation on system level. The exact data items and relationships that are to be used for creating networks have to be tailored to organisations depending on their way of working. For example, while some organisations might make heavy use of tracing between tests and requirements, in other organisations this kind of tracing could be missing.

LoCo CoCo relies exclusively on existing systems engineering data as the *ground truth*, in contrast to other approaches that either analyse actual communication [14, 13, 15] or manual recommendations [21, 22]. This means that data can be missing, e.g., not assigning a creator to an item; faulty, e.g., assigning the wrong person as a creator of an item; or contradicting, e.g., having two different creators for the same item in two different tools. In particular, social data are

often not a primary concern in systems engineering tools and, therefore, their quality can be low. However, our evaluation indicates that the data are good enough to render the networks useful for finding experts. While related approaches such as Codebook circumvent some of these issues, e.g., by analysing actual communication on an item, they suffer from similar limitations, as they use techniques such as natural language processing that can produce false positives. Furthermore, LoCo CoCo is intentionally designed to use data that are potentially incorrect, as any user of LoCo CoCo is presented with a network mirroring the data contained in systems engineering tools he/she uses. This means that, while low-quality data can cause networks that do not represent the actual or desirable communication channels, the visualisation can aid engineers to identify erroneous data in the systems engineering tool.

In our studied case, the extraction of relevant data and construction of any network took less than one minute. This means that social networks can be created on demand by practitioners in a much more efficient fashion than manual creation would allow.

Regarding RQ2, *How do practitioners evaluate the potential of these networks to tackle known communication challenges in systems engineering?*, practitioners indicated that it could be useful to tackle existing communication challenges, despite the low data quality. This indicates that the data quality is not the most important concern. Instead, we observe from the substantial improvement in the usefulness evaluation from the first to the second cycle that the concrete items and relationships that are used are of importance, but depend to a large extent on the organisation structure and an organisation's way of using different tools. Regarding the use of LoCo CoCo, we observe that practitioners have difficulties coming up with clear, precise use cases for social networks. Therefore, concrete use cases need to be provided when deploying LoCo CoCo or similar approaches in an organisation. This finding is in contrast to Codebook [15], which was built based on use cases provided in a developer survey. Possibly, this can be attributed to the fact that Codebook is rather focused on artefacts close to source code, which could be easier to grasp for many software engineers. Finally, networks that show connections between people across organisation boundaries are more useful, as practitioners are in many cases already aware of the network within their own group or department.

Our findings are not easily comparable to related publications, as those typically target different use cases, such as stakeholder identification for elicitation [21, 22], or focus on source code and developers [15, 16]. Similar to the findings in [21, 22], engineers clearly see the usefulness and could imagine using LoCo CoCo in practice. In contrast to [15, 16], we so far only conducted a static validation, without actually deploying LoCo CoCo in a complete project or organisation. Therefore, we could not collect any actual usage data from engineers. This is attributed to the fact that we conducted this study in an embedded systems context, where the development of new products can take 10 years or more. In this context, requirements and other artefacts that are implementation-independent evolve slowly. Hence, the actual usefulness of LoCo CoCo in practice can only be measured in a longitudinal study, which is left for future work. Instead, we

provide more in-depth insights from the feedback obtained during the evaluation.

While our interviewees had difficulties to provide additional use cases, we were able to gather a few in the second design science cycle, thus answering RQ3, *For which additional use cases would practitioners like to use LoCo CoCo?*. We identified two classes of use cases, namely those cases in which a user requires detailed information and those in which a user requires an overview. The former class of use cases aims at understanding the context of a particular part of the system, e.g., understanding the complexity of a component. The latter class aims at organisation issues, e.g., understanding where communication breakdowns might occur. This latter class is related to the findings in [16], where interviewees stated that it would be useful to see the current project activity based on code changes. Interestingly, in Begel et al.'s developer survey [15], no use case to provide an overview of the current situation, i.e., the latter class of use cases, can be found.

The final research question RQ4, *To what extent can LoCo CoCo be used with different tools and organisational contexts?*, can be answered as follows. So far, we implemented LoCo CoCo mainly for a single tool. However, we additionally provide a meta model for generalising the approach to other tools, evaluating this meta model with an additional tool. The results from this evaluation indicate that LoCo CoCo could be extended to a wide variety of tools. To lower the implementation effort, it would be preferable to implement adapters to existing tool interoperability standards like OSLC [40]. In this way, multiple tools could be connected to LoCo CoCo using a single implementation.

Technical concerns aside, it has to be noted that a certain amount of data is required for LoCo CoCo to be useful, as discussed for RQ1. We expect that in most cases where communication challenges occur, this amount of data is present. However, this needs to be verified in future work, especially with agile processes that focus on little documentation.

9. Conclusions and Implications

In this paper, we presented LoCo CoCo, the Low-Cost Communication and Coordination approach. In order to serve as a support tool for finding experts, LoCo CoCo constructs networks of people in an organisation using systems engineering data.

We constructed LoCo CoCo in a three-cycle design science study, evaluating it with a total of 15 interviews and 12 survey answers. Our results indicate that it is indeed possible to create social networks from existing systems engineering data. While social data are often not a primary concern, which can limit the quality of these data, our evaluation indicates that the data are good enough to render the networks useful for finding experts.

Our evaluation points to several key points to make LoCo CoCo or similar approaches successful in a company. First, practitioners seem to have difficulties coming up with clear, precise use cases for social networks. Therefore, even if

they have an abstract idea of the purpose of such an approach, concrete use cases need to be provided when deploying LoCo CoCo in an organisation. Secondly, networks that show connections between people that cross organisation boundaries are most useful, as practitioners are in many cases already aware of the network within their own group or department. Thirdly, explanations for edges between people need to be provided, e.g., the requirements that connect two persons. Finally, the concrete data to use are company-specific, as they depend on the company's processes.

To make LoCo CoCo applicable in a wide range of organisations, we constructed a tool-independent meta model for social networks. We then evaluated this meta model by incorporating data from two different systems engineering tools. While this attempt was successful, it should be evaluated with additional tools and in different organisations.

Finally, a number of ethical issues came up during our study. Importantly, these issues came up even though we only used data that were already existing and accessible at the case company. Clearly, this point needs to be studied in further detail in the future. One concrete solution to lower ethical concerns would be to implement LoCo CoCo as a query system, without visualising the entire network. However, this would also lower the usefulness of LoCo CoCo.

We are currently implementing the conceptual design as a customised version of Gephi[8], which will be released as open source.

Acknowledgement

We would like to thank our industry contacts, as well as all practitioners participating in the evaluation of LoCo CoCo. This study has been conducted as a master thesis at Chalmers University of Technology. In this context, the first author has received financial support from a Swedish Institute scholarship.

References

[1] D. E. Perry, N. A. Staudenmayer, L. G. Votta, People, organizations, and process improvement, IEEE Software 11 (4) (1994) 36–45. doi:10.1109/52.300082.

[2] R. E. Kraut, L. A. Streeter, Coordination in software development, Commun. ACM 38 (3) (1995) 69–81. doi:10.1145/203330.203345.

[3] J. D. Herbsleb, R. E. Grinter, Architectures, coordination, and distance: Conway's law and beyond, IEEE Software 16 (5) (1999) 63–70. doi:10.1109/52.795103.

[8] https://gephi.org

[4] 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.

[5] B. Curtis, H. Krasner, N. Iscoe, A field study of the software design process for large systems, Commun. ACM 31 (11) (1988) 1268–1287. doi:10.1145/50087.50089.

[6] M. Niazi, S. Mahmood, M. Alshayeb, M. R. Riaz, K. Faisal, N. Cerpa, S. U. Khan, I. Richardson, Challenges of project management in global software development: A client-vendor analysis, Information and Software Technology 80 (2016) 1–19.

[7] J. D. Herbsleb, Global software engineering: The future of socio-technical coordination, in: Future of Software Engineering, 2007. FOSE '07, 2007, pp. 188–198. doi:10.1109/FOSE.2007.11.

[8] G. Liebel, M. Tichy, E. Knauss, O. Ljungkrantz, G. Stieglbauer, Organisation and communication problems in automotive requirements engineering, Requirements Engineering (2016) 1–23doi:10.1007/s00766-016-0261-7.

[9] 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.

[10] J. Srivastava, Data mining for social network analysis, Intelligence and Security Informatics (2008) 33–34.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. Begel, Y. P. Khoo, T. Zimmermann, Codebook: Discovering and exploiting relationships in software repositories, in: Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1, 2010, pp. 125–134. doi:10.1145/1806799.1806821.

[16] A. Mockus, J. D. Herbsleb, Expertise browser: a quantitative approach to identifying expertise, in: 24th international conference on software engineering (ICSE '02), 2002, pp. 503–512.

[17] EAST-ADL Association, EAST-ADL v2.1.12, http://www.east-adl.info/Specification/V2.1.12/html/index.html, last accessed Jun. 2015.

[18] AUTOSAR, AUTOSAR, http://www.autosar.org/, last accessed Aug. 2015.

[19] A. Begel, N. Nagappan, C. Poile, L. Layman, Coordination in large-scale software teams, in: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, IEEE Computer Society, 2009, pp. 1–7. doi:10.1109/CHASE.2009.5071401.

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

[21] S. L. Lim, D. Quercia, A. Finkelstein, Stakesource: Harnessing the power of crowdsourcing and social networks in stakeholder analysis, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, 2010, pp. 239–242.

[22] S. L. Lim, D. Damian, A. Finkelstein, Stakesource2.0: Using social networks of stakeholders to identify and prioritise requirements, in: Proceedings of the 33rd International Conference on Software Engineering, 2011, pp. 1022–1024.

[23] A. Guzzi, A. Begel, Facilitating communication between engineers with cares, in: 34th International Conference on Software Engineering (ICSE '12), 2012, pp. 1367–1370.

[24] A. Guzzi, A. Begel, J. K. Miller, K. Nareddy, Facilitating enterprise software developer communication with cares, in: 28th IEEE International Conference on Software Maintenance (ICSM '12), 2012, pp. 527–536.

[25] T. Gorschek, P. Garre, S. Larsson, C. Wohlin, A model for technology transfer in practice, IEEE Software 23 (6) (2006) 88–95. doi:10.1109/MS.2006.147.

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

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

[28] L. McLeod, S. G. MacDonell, B. Doolin, Qualitative research on software development: a longitudinal case study methodology, Empirical Software Engineering 16 (4) (2011) 430–459. doi:10.1007/s10664-010-9153-5.

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

[30] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, Experimentation in Software Engineering, Springer, 2012.

[31] A. Fink, The survey handbook, Vol. 1, Sage, 2003.

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

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

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

[35] A. Zaczek, D. Schmitt, Graph#, https://graphsharp.codeplex.com/, last accessed Apr. 2016.

[36] A. Cooper, R. Reimann, D. Cronin, C. Noessel, About Face: The Essentials of Interaction Design, Wiley, 2014.

[37] C. Ware, Information Visualization: Perception for Design, Information Visualization: Perception for Design, Morgan Kaufmann, 2013.

[38] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Computers in human behavior 51 (2015) 915–929.

[39] K. Dikert, M. Paasivaara, C. Lassenius, Challenges and success factors for large-scale agile transformations: A systematic literature review, Journal of Systems and Software 119 (2016) 87–108.

[40] OSLC, Open Services for Lifecycle Collaboration, https://open-services.net/, last accessed Jun. 2017.

Appendix A. Data Collection Instruments

Appendix A.1. Interview Guide Cycle I

1. How do you rate the accuracy of LoCo CoCo based on the supplied networks? (Likert, 1 to 10)
2. What data should we add in order to improve the accuracy? (Free text)
3. How useful do you think this approach is for finding experts for clarifications? (Likert, 1 to 10)
4. Are there any additional use cases you could see for LoCo CoCo? (Free text)

Appendix A.2. Interview Guide Cycle II

The interviews in the second cycle were unstructured and loosely followed the interview guide from the first cycle.

Appendix A.3. Survey Cycle II

1. In which department do you work? (Free text)
2. What is your role in the department? (Free text)
3. How long have you worked for the company? ($<$ 6 months, 6 months to 2 years, 2 years to 4 years, $>$ 4 years)
4. How long have you worked for the department? ($<$ 6 months, 6 months to 2 years, 2 years to 4 years, $>$ 4 years)
5. How often do you use SystemWeaver? (daily, weekly, monthly, no active use, never)
6. How useful is this approach of building social networks as a whole? (Likert, 1 to 10)
7. I would use this kind of approach to (multiple possible): Find the right person to talk to, Find 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, Other (Free text)
8. What limitations does this approach have? (Free text)
9. What additional features may make the approach more useful? (Free text)
10. For each of six implemented sub features:
    (a) How accurate are the networks (for the sub feature)? (Likert, 1 to 10)
    (b) How useful are the networks (for the sub feature)? (Likert, 1 to 10)
    (c) What are the limitations of the sub feature? (Free text)
    (d) What possible (additional) use cases does this sub feature offer for your work? (Free text)

Appendix A.4. Interview Guide Cycle III

1. What tools do you usually use at work?
    (a) Do these tools hold social data?
    (b) Are these tools connected (Do they reference each other)?
    (c) Are the tools models transformable to the presented integration model?
    (d) If no, why not and is it possible to alter the model so that the transformation becomes possible?
2. What is your general impression on this generalisation model and how can we improve it?
3. How do you find this compared to the old visualisation with regards to your use cases?
4. By looking at the colours and shapes of the persons nodes, do you feel that some people are more important than other?
5. In your opinion, how can this tool be misused?
6. How do you feel to be a part of a network?
    (a) How do you feel if your node is isolated?
    (b) How do you feel if your node is heavily connected?
    (c) Do you feel OK for your information to appear in the application (your picture included)?