International Journal of Software Engineering
and Knowledge Engineering
Vol. 29, No. 10 (2019) 1497–1538

DOI: 10.1142/S0218194019500475

World Scientific
www.worldscientific.com

Communication Problems in Software Development — A Model and Its Industrial Application

Joakim Pernstål
Volvo Car Corporation, SE-405 31 Göteborg, Sweden
jpernsta@volvocars.com

R. Feldt
Chalmers University of Technology, SE-412, 90 Gothenburg, Sweden
robert.feldt@chalmers.se

T. Gorschek
Blekinge Institute of Technology
SE-372 25 Karlskrona, Sweden
tony.gorschek@bth.se

D. Florén
Volvo Car Corporation, SE-405 31 Göteborg, Sweden
dfloren@volvocars.com

Received 2 January 2019
Revised 27 February 2019
Accepted 8 April 2019

Attaining effective communication within and across organizational units is among the most critical challenges for success in software development organizations. This paper presents a novel model, supporting analysis of problems in inter-departmental communication events. The model was developed and designed based on industrial needs emphasizing flexibility, applicability and scalability. The model covers central communication aspects in order to provide a useful approximation of communication problems rather than in-depth modeling on message-by message basis. Other event-specific information, such as costs, can then be attached to enrich analysis and understanding. To exemplify and evaluate the model and collect feedback from industry, it was applied to 16 events at a Swedish automotive manufacturer where communication between two departments had broken down during development of software-intensive systems. The evaluation showed that the model helped structure and conduct systematic data collection and analysis of dysfunctional communication patterns. We found that insufficient understanding of the matters being communicated was prevalent, but also more specifically, requirements were insufficiently balanced, detailed and specified over the full system development cycle. Besides, the long-term cost for the company was analyzed in depth for each event, yielding a total estimated cost for the analyzed communication events of 11.2MUS$.

Keywords: Organizational management and coordination; communication; software engineering; manufacturing engineering; software-intensive systems; automotive industry.

1. Introduction

Software is becoming an increasingly important component in traditionally hardware-intensive industries (e.g. automotive and aerospace) [1, 2]. In these organizations, but generally in large organizations, the software-intensive systems are commonly developed in the context of large-scale system development (i.e. systems of systems development), where software constitutes only one, but important, part of the whole [1–3]. With the development of software-intensive systems follows many challenges, among which coordination and communication between different competencies and departments are central.

For minimizing the complexity and amount of communication performed between teams in development work, the modularization approach has been promoted [4, 5]. However, in many products (e.g. vehicles) the amount of interacting software-intensive systems has increased dramatically, challenging the possibilities to modularize, therefore prompting the need for communication [1]. Using an automotive car manufacturer as a case, coordination and communication between individuals and groups (e.g. system owners, architects and developers) is critical not only within a department (intra-departmental issues), but also across departments (inter-departmental issues). This paper and the communication model presented focuses on the latter — inter-departmental communication between Product Development (PD) and Manufacturing (MAN) — as it has been identified as challenging in PD [6–9], and in particular, when it comes to development of software-intensive automotive systems [10]. The inter-departmental coordination and communication in our definition includes all the phases of design and development, from concept (e.g. exploration and balancing of requirements and solutions) to design, implementation and validation, but also manufacturing involving pre-production verification and validation of the manufacturing systems cf. [10–13]. In a new car model project, these activities commonly span over 2–4 years [11, 12].

Both the literature in organizational theory and research on large-scale software development projects suggest that coordination and communication across organizational boundaries is a key factor for success as the level of complexity, uncertainty and interdependency of the work performed by an organization increases [14–16]. This is also stressed in development methods based on agile and lean approaches (e.g. Scrum [17] and lean software development (LSD) [18].

Several empirical studies have been reported on communication in large-scale software development (e.g. [15, 16, 19, 20]). Mainly focusing on requirements communication, some of the previous work has analyzed data by using different analytical models and concepts [21–28].

The main purpose of this paper is to advance a communication model, helping researchers and practitioners to empirically investigate organizational communication problems on different levels in large-scale software development efforts. With the underlying goal of revealing effects and causes on which solutions can be developed, our model structures, visualizes and classifies the main characteristics of

communication problems by examining whole series of messages and interactions that have occurred during different organizational communication events in concluded projects.

A tangible industrial need, motivated our efforts to develop an analytic tool called Software Communication Redundancy and Effectiveness Analysis Model (SCREAM). We developed and used the model in close collaboration with a Swedish automotive company, namely Volvo Cars. To demonstrate how to apply the model in practice, and evaluate the practical usefulness of the model we conducted a case study, in which we selected and then analyzed in depth a set of 16 real communication events at Volvo Cars. The events represented different matters of concern where the interplay between PD and MAN in development of software-intensive automotive systems was central. PD is concerned with design and development of software-intensive automotive systems (e.g. development of powertrain and chassis control systems for vehicles). MAN is concerned with managing these systems when producing vehicles (e.g. vehicle manufacturing operations affected by powertrain and chassis control systems). The communication events were identified through conducting meetings with key informants and reviewing archival data at Volvo Cars.

The paper is organized as follows. Section 2 gives an overview of related work. Section 3 describes and exemplifies the design of SCREAM and Sec. 4 presents the research context and how the model was applied. The main results of using the SCREAM are presented and analyzed in Sec. 5. The results and limitations of our work are discussed in Sec. 6. Finally, conclusions and future work are presented in Sec. 7.

2. Related Work

This section gives an overview of communication theory, focusing on communication models that are relevant for our work.

2.1. *Communication theory*

Communication is a broad concept and even though many theories and models for it have been proposed it has been argued that it cannot be considered a single field [29]. In the following, we give a brief overview of the most related work. Additional information can be found in textbooks such as [30, 31].

Several models originating from the traditional sender/receiver model of communication and associated concepts, have been suggested for describing and analyzing the communication process (e.g. [32–35]). In general, most models include or cover at least the elements of communication, sender, receiver, medium and message, but there is no model that considers all aspects of communication as it would be too complex and detailed [36, 37]. Additional elements are, for example, noise source — disturbance changing the original message from the sender and context — social situation in which the communication is taking place.

Clark and Schaefer [38] suggest an extension of the traditional sender/receiver model of communication, building on the notion of "common ground" and "grounding" (a.k.a. the contribution theory). The contribution theory extends the view of single messages to an analytic frame where contributions (utterances) are jointly developed and produced in order to achieve a common understanding. A further development of the contribution theory is proposed by Clark and Brennan [39], where two key factors impacting the grounding process: purpose and medium, are discussed. They also suggest that the production of contributions to conversations have two phases: presentation phase and acceptance phase, where the receiver's level of understanding can be divided into four states, States 0–3, after the presentation phase. In State 3, the sender and receiver have achieved a common understanding, i.e. a common ground has been accomplished.

In organization theory, the information processing theory and organizational design models have been frequently used to analyze and describe boundaries in PD (e.g. [40, 41]. Based on the information processing theory and the social presence theory, the media richness theory is presented in [42–44]. It proposes that the efficiency of organizational communication is affected by the match between the media richness and the characteristics (e.g. complexity and ambiguity) of the task being communicated. The theory hypothesizes that using richer media (e.g. face-to-face) for more complex tasks has a positive effect on communication. Clark and Brennan [39] also emphasize the importance of appropriate media selection and discuss the impact of different communication technologies' constraints and resources on the process for producing information to a shared understanding. In large-scale software projects, the value of combining both formal communication (e.g. written and transferred specifications and structured meetings) and informal communication (e.g. unscheduled face-to-face meetings and e-mail or phone conversations) within and across departments has been stressed [14]. Tests of the media richness theory have shown positive results but also limitations [28, 45]. For example, in requirements engineering (RE), Calefato *et al.* [28] found that text-based communication was preferred when having open discussions on conflicting issues and there was no significant difference between the use of face-to-face and text-based medium in requirements elicitation concerning satisfaction with performance.

Social network theories have been used for studying organizational communication as they provide a way to visualize and analyze patterns of relational and structural perspective [46, 47]. In network analysis, there are a number of measures such as density and centrality that are helpful in examining networks. Informal network structures have been seen as being more worthy of study than formal ones because they are seen as promoting a better understanding of organizational behavior [48, 49]. Using theories on network analysis, the impact of a given network structure on organizations' capability to share knowledge, norms and behaviors across organizational boundaries have also been investigated [50, 51]. Recently, analytic tools adopting a social network analysis approach using data from software repositories has been promoted to investigate the impact

of communication patterns (CPs) on the performance of software development teams in industry [52, 53].

Division of labor occurs prior to coordination, where development work is divided into tasks and sub-units (individuals, groups), and departments are assigned to each task [54]. Achieving effective coordination and communication within and between workgroups have been acknowledge in both research and industry as a critical success factor in software development projects [15, 16, 55]. To reduce the interdependencies between tasks, the idea of modularization has been promoted, particularly in systems design and software engineering (SE) [4, 5]. It builds on the observation that the product architecture reflects the organizational structure (Conway's law) and is a useful approach for dividing the development of complex products into independent and manageable development and manufacturing tasks (e.g. [56, 57]).

Building on Conway's law, Cataldo *et al.* [21] propose the socio-technical congruence (STC) model. STC provides a fine-grained view of required coordination needs based on the technical and social relationships, indicating alignment between coordination needs and actual coordination. Any misalignments are identified as socio-technical gaps.

A number of empirical inquires have studied the effects of the concept of congruence on software teams in industry. Cataldo *et al.* [58] observed that higher congruence leads to faster completion of modification requests and that socio-technical gaps have a negative effect on productivity, which is in line with [59]. Furthermore, Cataldo *et al.* [58] also found that the coordination needs across team boundaries were constantly and substantially changed over time.

Kwan *et al.* [26] applied an unweighted and a weighted (considers strength of relationships) STC approach, as described in Kwan *et al.* [25], to a data set containing 191 concluded software builds over one-year period in a large software project. They found that increased congruence has a positive effect on the results of projects classified as continuous build (includes components changes from the local development site and stable components from remote sites). However, in software projects where new components from every development site need to be integrated (integration build) increased congruence has a negative effect on the project success. Another finding was that the software builds were successfully accomplished even though low congruence values. They observed that one reason for this was modular design but also a strong awareness of each other's work among the team members primarily through informal communication.

In a case study, Kwan and Damian [24] identified mechanisms used for enabling software team members to absorb awareness information. Most of these mechanisms were based on simple communication techniques. In addition, they observed that experienced team members acted as brokers to bridge coordination gaps. Overall, they concluded that the STC approach was not sufficient for examining some of the situations observed. As a consequence, the STC approach was extended by

incorporating both awareness and brokerage. To the best of the authors' knowledge, however, it has not been validated.

Similarly, Marczak *et al.* [22] elaborated a model based on STC for studying the communication of requirements changes across teams, and in particular, the impact of brokers on the information flow. The model was applied to pairs of interdependent requirements, which is the simplest case, and it is unclear whether the model is scalable for larger projects involving interdependencies between multiple requirements, teams and departments.

Overall, studies on STC models show that they provide a useful way to better understand coordination problems in industry but they also have limitations. The models only provide a snap-shot of the communication situation, they are context sensitive making it difficult to generalize their applicability and results. Even though nothing prevents STC models to analyze inter-departmental communication, no empirical studies demonstrating their effects on such settings in industry have been found, and their scalability are unclear. Furthermore, the type of information exchanged and whether the content of the information was transferred correctly is not exhibited.

Inspired by the concept of the goal-oriented systems theory, as described in Klir [60], Fricker *et al.* [27] developed a goal-oriented requirements communication model. Primarily, the model serve as reference for analyzing to what the degree of a bidirectional communication between the two communicating actors has been established. Basing on the model, Fricker *et al.* [61] successfully applied a solution for requirement negotiation between program managers and development teams where the roles of goal seeker and implementer are rather clear and static over time. However, the roles the actors play may vary depending on factors such as the reason for the communication and what is communicated, and can shift during the development cycle (see Sec. 3). Furthermore, the main content of the interaction can be other development artifacts than requirements, such as functional specifications and models, and code, for which the model has not yet been demonstrated. Moreover, the model does not provide recommendations for what information about the interactions investigated is central to collect and how this information should be collected and categorized in order to reveal causes and effects of communication failures.

2.2. *Summary of communication theories and models*

In summary, most of the earlier work is limited to intra-departmental communication of requirements and do not cover other development artifacts, and if the communication is being modeled the applicability of the models to inter-departmental communication in industry is unclear. Table 1 gives a summary of communication theories and models as described in Sec. 2.1 and SCREAM (see Sec. 3), and their main features in relation to the most common elements of communication [36, 37].

Table 1. Summary of communication theories and models.

	Elements of communication				
Theory/Model	Sender (who is sending the message?)	Receiver (who is intended recipient of the message?)	Message (what is being communicated?)	Medium (how is the message communicated?)	Effect (the result of the communication)
Sender/receive models [34]	Transmits (codes) a signal suitable for the message	Reconstructs (decodes) the message from the signal	Information being communicated on message by message basis	The connection between the sender and receiver	Messages are communicated between the parties with a shared code
Contribution theory e.g. [38, 39]	Produces meaningful information in collaboration with the receiver	Actively involved in establishing meaning of the information sent	Single messages (utterances) are developed to meaningful contributions	Capability to attain common understanding	The level of a common understanding attained
Media richness [42–44]	Organizational entity producing and sending information	Organizational entity receiving information	Series of messages conveyed and their characteristics concerning variety (predict problems) and analyzability (respond to problems)	Focus on the richness of media that varies depending on feedback, multiple cues, language variety, and personal focus	Match between the media richness and the characteristics of the information being communicated
Social network [47, 48]	Node positioned in the network representing a social actor	Node positioned in the network representing a social actor	Series of messages or resources flowing through the ties	Ties connecting the nodes and representing relationships/interactions	Impact of the network's pattern on communication, concerning, for example, centrality and density
STC [21]	Actors and their assignments to work tasks	Actors and their assignments to work tasks	Series of messages related to the work tasks conducted	Capability to meet the coordination needs of work tasks based on the characteristics of the technical entities and their dependencies	Alignment between coordination needs and actual coordination

Table 1. (*Continued*)

	Elements of communication				
Theory/Model	Sender (who is sending the message?)	Receiver (who is intended recipient of the message?)	Message (what is being communicated?)	Medium (how is the message communicated?)	Effect (the result of the communication)
Goal oriented communication [27]	Goal seeking element	Goal implementing element	Series of messages focusing on requirements specifications and design solutions	Facilitates requirements analysis and negotiation (e.g. review and sign-off meetings)	Optimized using four paradigms: informationless (the weakest paradigm), feed forward, feedback, and full-information (the strongest paradigm)
SCREAM (Sec. 3)	Actor including two attributes representing the level of understanding and the actions taken	Actor including two attributes representing the level of understanding and the actions taken	Series of messages focusing on PD artifacts	Relevance and completeness of the specifications clarifying the exchanged information	The relevance and establishments of communication and specifications, the level of a common understanding, and the actions taken

3. Communication Model

For industrial applicability, it is important that models and analytical tools are easy to grasp and focus on the central aspects being modeled [62]. Central to any communication are two nodes, typically modeled as a sender and a receiver, and a message that is to be transmitted between them to reduce uncertainty and create a shared understanding [34]. In later work, communication frameworks based on the contribution theory, elaborating on extensions of the sender-receiver turn-taking view have been suggested [38, 39]. In general, they see communication as a collaborative process-labeled grounding where the participants jointly develop and specify exchanged information (contributions) in order to achieve a shared understanding.

The development of SCREAM was conducted in collaboration with Volvo Cars in series of workshops where key company representatives participated and contributed with their expertise and knowledge to various communication issues that were raised and discussed. For example, how much communication was needed and if it has actually happened, and if the communicating actors have understood the matter being communicated and have acted on it. A key issue in software development is also to what degree a decision or related information needs to and has been specified and refined — it is well known that much information is tacit and never written down [16].

For simplicity, and to increase applicability, we decided to focus on binary levels for each modeling element, at least as a first approximation. We considered many possible modeling elements, such as communication channel and medium, effects, and number and types of actors (e.g. external actors or mediators affecting the results), but after several workshops and iterations four main elements were selected, each with two attributes that are both binary-valued. Additional information can then be collected for each communication event being modeled and attached to the modeled event.

In Fig. 1, the resulting layout of SCREAM is shown. It consists of four main elements: sender, receiver, communication (C), and specification (S). Each element has two attributes used to describe different communication events in an organization. Our main intention of SCREAM is that it should be capable of postmortem modeling of communication events, where weak communication between two actors has occurred and capturing relevant meta-data associated with the event examined. Furthermore, to attain a general model, it should be possible to tailor for communication in organizations facing other types of business and settings.

Depending on the characteristics of the event examined, the communicating actors can either be represented by the sender or the receiver element in SCREAM. Our definition of the sender is the actor who starts the communication by sending the initial message and the receiver is the actor that is expected to respond to the initial message from the sender. Depending on the level of analysis (e.g. [15]), the sender and receiver can be individuals, roles, groups, teams, projects, and organizations. For example, updated requirements or specifications on car functions are owned by the

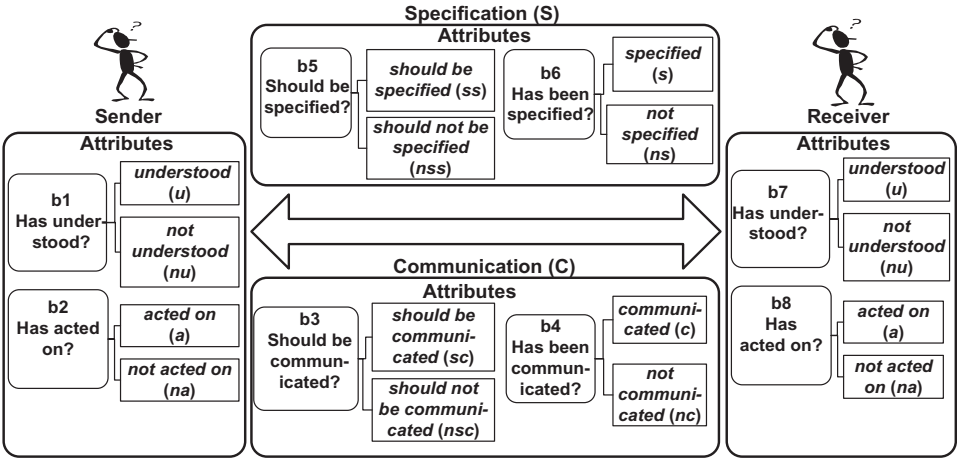

Fig. 1. The layout of SCREAM and its elements.

function owners and initially sent from them to intended receivers (e.g. product managers, system owners and design engineers) for review and acceptance.

Different patterns of communication can be modeled by chaining the elements and the coding of their attributes. We define the CP as a modeled representation of the communication event investigated, and is coded in a binary string, as expressed in (1).

$$\begin{aligned} &\text{CP} := \text{sender } (b1, b2); C(b3, b4); S(b5, b6); \text{receiver } (b7, b8), \\ &b1, b7 \in \{u, \text{nu}\}; b2, b8 \in \{a, \text{na}\}; b3 \in \{\text{sc}, \text{nsc}\}; b4 \in \{c, \text{nc}\}, \\ &b5 \in \{\text{ss}, \text{nss}\}; b6 \in \{s, \text{ns}\}. \end{aligned} \tag{1}$$

Given the restriction that each attribute can have two states (e.g. u = understood or nu = not understood for $b1$ and $b7$) when analyzing a communication event between two actors (e.g. system and software designer or product manager and function developer), SCREAM can theoretically represent $2^9 = 512$ different CPs. However, the value of 512 is only a theoretical upper limit, since some representations may not be realistic in practice. For example, identifying representations including the combination C(nsc, nc)S(nss, ns) are not likely — traces of unexpected and not established communication and specification are difficult to find. This reduces the number of realistically possible CPs with $2^5 = 32$ to 480 (512–32). The following describes and exemplifies the coding of the elements' attributes.

The sender element contains two attributes. The first attribute is mainly based on the notion of common ground and grounding, giving a binary representation of the states of the level of understanding that has been achieved as posited in [39]. Accordingly, the value nu represents States 0–2 while the value u represents State 3. However, reaching a common understanding does not mean that the actors have taken any actions affecting each other. Therefore, a second attribute was included to

model if the sender has taken any actions on the matter, affecting the receiver, except actions related to the communication of the matter. For example, if the system designer has acted on the impact of increasing the functions of the system by, for example, adapting the memory size and computational capacity (e.g. Flops) of the system, then the system designer has acted on the matter. But if the system designer has sent function and system requirements to the software designer, then the matter is attributed as communicated. Table 2 describes and exemplifies the sender attributes.

Table 2. Sender attributes.

	Has acted on?	
Has understood?	Acted on (a)	Not acted on (na)
Understood (u)	The sender has understood the matter and has acted on it. Example System designer (u, a): The system designer has **understood** the impact of increasing the functions of the system. The system designer has **acted** by adapting the memory size and computational capacity of the system.	The sender has not understood the matter and has not acted on it. Example System designer (u,na): The system designer has **understood** the impact of increasing the functions of the system The system designer has **not acted** by adapting the memory size and computational capacity of the system.
Not understood (nu)	The sender has understood the matter, but has not acted on it. Example System designer (nu,a): The system designer has **not understood** the impact of increasing the functions of the system. The system designer has **acted** by changing the system design but without considering required changes of memory size and computational capacity.	The sender has not understood the matter, but has acted on it. Example System designer (nu,na): The system designer has **not understood** the impact of increasing the functions of the system. The system designer has **not acted** by changing the system design.

Table 3 describes and exemplifies the receiver attributes. Similar to the sender attributes, the receiver attributes model the receiver's capability and willingness of understanding and assimilating the sender's information, and whether the receiver has taken any related actions on the matter.

The communication attributes provide information about whether the matter of concern should and has been communicated between the sender and the receiver. Table 3 describes and exemplifies the communication attributes.

According to Clark and Brennan [39], specifying the exchanged information is an important part of the grounding process to effectively reach a shared understanding. The attributes of the specification element reflect the relevance and completeness of the exchanged information specified, and whether this has been carried out, so the actors can effectively and sufficiently understand and absorb it. Table 4 describes and exemplifies the coding of these attributes.

Table 3. Receiver attributes.

	Has acted on?	
Has understood?	Acted on (a)	Not acted on (na)
Understood (u)	The receiver has understood the message from the sender and has acted on it. Example Software designer (u,a): The software designer has **understood** the impact of increasing the functions of the system on the software design. The software designer has **acted** by adapting the software design so its impact on memory size and computational capacity are optimized.	The receiver has understood the message from the sender, but has not acted on it. Example Software designer (u,na): The software designer has **understood** the impact of increasing the functions of the system on the software design. The software designer has **not acted** by adapting the software design even though seeing its impact on memory size and computational capacity.
Not understood (nu)	The receiver has not understood the message, but has acted on it. Example Software designer (nu,a): The software designer has **not understood** the impact of increasing the functions of the system on the software design. The software designer has **acted** by changing the software design but without seeing the impact on memory size and computational capacity.	The receiver has not understood message from the sender, and has not acted on it. Example Software designer (nu,na): The software designer has **not understood** the impact of increasing the functions of the system on the software design. The software designer has **not acted** by changing the software design.

The following example of a communication event shows how it can be modeled as a CP by chaining the four elements in SCREAM and coding their attributes, and what kind of information the CP can provide for further analysis. In this example, the system designer has understood the impact of adding functions to the system and acted by adapting the memory size and computational capacity of the system.

The system designer should and has initiated the communication by sending function and system requirements to the software designer and the software designer should respond on the requirements from the system designer, i.e. the system designer act as the sender and software designer act as the receiver. The system designer and the software designer have been involved in the specification of the adaptions between the system and software design but the adaptations between the system and software design solutions have not been sufficiently specified. So, the software designer misunderstood some of the requirements and acted by changing the software design but without understanding its impact on the system solution for memory size and computational capacity. This communication event can be represented as the CP in (2).

$$\text{CP} := \text{System designer}(u, a); C(\text{sc}, c); S(\text{ss}, \text{ns}); \text{Software designer}(\text{nu}, a). \quad (2)$$

The coding of the CP shows that even though the communication between the system designer and the software designer was established and the software designer

Table 4. Communication attributes.

Should be communicated?	Has been communicated?	
	Communicated (*c*)	Not communicated (nc)
Should be communicated (sc)	The matter should and has been communicated between the sender and receiver. Example C (sc,*c*): **Should be communicated** to the software designer since increasing the functions of the system has an impact on the software design. The system designer has **communicated** new function and system requirements to the software designer.	The matter should, but has not, been communicated between the sender and receiver Example C (sc,nc): **Should be communicated** to the software designer since increasing the functions of the system has an impact on the software design. The system designer has **not communicated** new function and system requirements to the software designer.
Should not be communicated (nsc)	The matter should not, but has been communicated between the sender and receiver. Example C (nsc,*c*): **Should not be communicated** to the software designer since increasing the functions of the system concerns only hardware and not the software design. The system designer **has communicated** irrelevant function requirements to the software designer which has started unnecessary information exchange, i.e. redundancy.	The matter should not, and has not, been communicated between the sender and receiver. Example C (nsc,nc): **Should not be communicated** to the software designer since increasing the functions of the system concerns only hardware and not the software design. The system designer **has not communicated** irrelevant functions requirements to the software designer.

has taken actions, the software designer did not gain sufficient understanding, and thus made the wrong assumptions when adapting the software design to the changes of the system solution. The CP indicates that a probable underlying cause for the failure of the communication can be related to specifications, clarifying the content of the exchanged information. Furthermore, root cause analysis by using collected meta-data of the communication event may reveal, for example, that too little effort was spent on specifying, or the overall specification processes and practices across the actors need to be improved.

In this section, we advanced a model called SCREAM that was developed in close collaboration with Volvo Cars and is mainly based on theoretical communication frameworks relying on the traditional sender/receiver model (e.g. [34]) and theories associated with the concept of common ground [38, 39]. Its overall goal is to be conceptually simple and practically useful and tailorable to suit different businesses and industrial settings. SCREAM also characterizes central aspects of organizational communication problems in a structured and descriptive way, emphasizing to reveal patterns of communication issues on a high-level and their effects and causes on which efforts for developing improvements can be motivated and based.

Table 5. Specification attributes.

Should be specified?	Has been specified?	
	Specified (*s*)	Not specified (ns)
Should be specified (ss)	The matter should and has been specified. Example S (ss,*s*): **Should be specified** since increasing the functions of the system has an impact on the software design. The system and software designers have been involved and **specified** the adaptations between the system and software design	The matter should, but has not, been specified. Example S (ss,ns): **Should be specified** since increasing the functions of the system has an impact on the system and software design. However, the adaptions between the system and software design were **not specified**.
Should not be specified (nss)	The matter should not, but has, been specified. Example S (nss,*s*): **Should not be specified** since increasing the functions of the system has no impact on the software design. The system and software designers have been involved and **specified** the adaptations between the system and software design, even though they should not, i.e. redundancy.	The matter should not, and has not, been specified. Example S (nss,ns): **Should not be specified** since increasing functions of the system has no impact on the software design. Therefore, the adaptations between the system and software design were **not specified**.

SCREAM includes four main elements: sender, receiver, communication, and specification. Each element has two attributes that are binary-valued and used to describe different communication problems in an organization. Different patterns of communication can be modeled by chaining the elements and the coding of their attributes into a binary string.

4. Industrial Application of Model

The work presented in this paper is based on empirical research primarily using a case study design, and qualitative and quantitative data [63, 64]. The flexible nature of this approach was found useful, since the overall objectives of this study are: (1) to iteratively develop and improve the design of SCREAM based on our experiences and through feedback from professionals on it while collecting and analyzing data for the 16 communication events and (2) to identify the main characteristics and develop an in-depth understanding about the communication problems arising in the case investigated and use this knowledge to support our industrial partner in creating and deciding on suitable solutions.

The study presented in this paper is part of a software process improvement (SPI) initiative focusing on the inter-departmental interaction between MAN and PD in development of software-intensive automotive systems. The results of the process assessment showed that many of the issues (e.g. RE and early manufacturing involvement) was related to insufficient communication [10].

The most critical issue revolved around the hand-shaking of requirements and further refinement and detailing of them along with managing changes throughout the full car development cycle. In our continued work with Volvo Cars, other types of communication situations between PD and MAN also surfaced. For example, problem solving and development and agreement on feasible solutions where communication of other development artifacts rather than requirements were central (e.g. concepts and functional specifications, models, and test cases). In addition, the cost for adapting the in-vehicle software to the manufacturing processes at Volvo Cars could be estimated to be between 2MUS$ and 8.5MUS$, depending on the degree of change in a new car model [10]. To gain a better understanding and improving the communication problems between PD and MAN, Volvo Cars expressed a direct need for examining and clarifying the complexity and uncovering the effectiveness of organizational communication in practice. This motivated our efforts to develop SCREAM, for post-mortem modeling and analysis of organizational communication problems.

Below we describe the industrial setting and present how SCREAM was applied. The following descriptions and examples are based on four of the 16 events: (1) calibrating the automatic opening-and-closing function of the tailgate system in manufacturing, (2) configuring the driver door system (DDS) by downloading software to the DDS Electrical Control Unit (ECU) in manufacturing, and (3) testing and verifying the audio system in manufacturing, and (4) verifying and documenting part and serial numbers in manufacturing for an infotainment component. This because Volvo Cars only allowed us to expose four events and we do not have room for the other 12 events.

4.1. *Industrial setting at Volvo Cars*

The study presented in this paper was carried out at Volvo Cars. The company has a long history of developing vehicles, and its internal culture is influenced by the Swedish cultural heritage. Volvo Cars is a premium car manufacturing company and has approximately 30,000 employees all over the world and produces roughly 500,000 cars per year (Volvo Cars 2016).

Volvo Cars develops the software-intensive systems in large-scale and costly projects, involving many people who represent several parts of the organization and engineering disciplines. The complexity of the cars is high, since they are built of interacting functions, systems, and sub-systems running on a large amount of software requiring extensive administration of requirements, specifications and standards and precise integration (over 300 systems, ∼2500 functions and about 100,000 textual requirements in Volvo Cars). Furthermore, Volvo Cars has operations in their manufacturing processes that are directly affected by in-vehicle software and must be considered during the development. For example, car configurations (e.g. assembly plant software download), and verifications (e.g. electrical tests) [10]. Also, the design and development activities span over 2–4 years for a new car model depending on the level of complexity and differentiation. This increases the

uncertainty because software's functionality is highly changeable over time and the tremendous growth of new software-intensive car functions [1, 15]. Given these characteristics for Volvo Cars, the need of inter-departmental communication and coordination throughout the development cycle is paramount [16].

The development of software-intensive systems is guided by the *V* model [65]. The deliveries of the *V* model are governed by the overarching development system for developing and delivering the complete vehicles based on a stage-gate model with milestones (MSs) [66, 67]. At Volvo Cars, the decision on project start is made at MS1. Exploration and balancing of requirements and solutions, and the status of specifications, development, integration, and verification for systems and component are reported on different levels (e.g. systems and components) at MS2 and MS3. Complete car prototypes are verified at MS4. Trial production and verification of the manufacturing processes starts at MS5, and manufacturing readiness and ramp-up of manufacturing are decided at MS6. Similar development approaches are commonly used in the automotive industry [1, 68].

4.2. *Applying SCREAM at Volvo Cars*

The application of SCREAM at Volvo Cars comprised three main steps as described in Fig. 2. The steps were performed iteratively over a period of three months.

4.2.1. *Step 1—Data collection*

In Step 1, the data were collected by using an approach similar to the critical incidents technique [69]. This technique can broaden knowledge of sparsely documented or

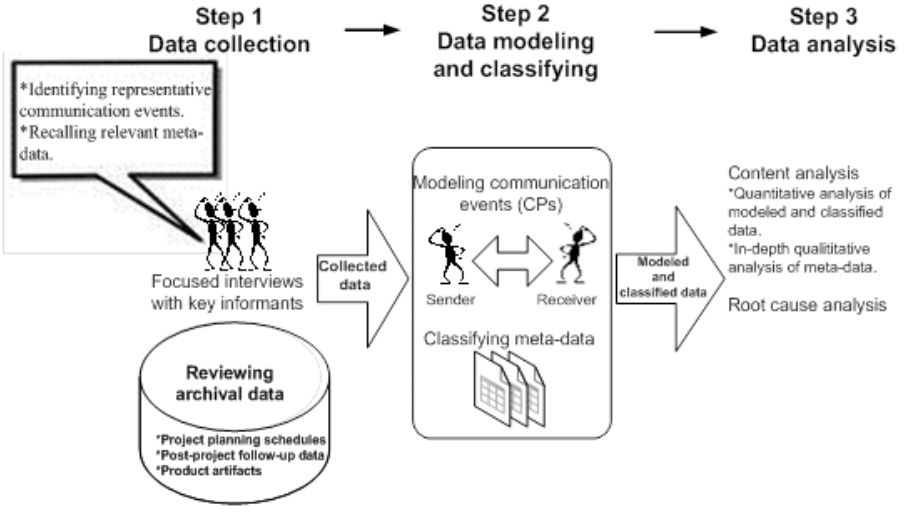

Fig. 2. Overview of applying SCREAM at Volvo Cars.

poorly understood areas using factual reports of an individual's observation of their behavior or of others [69, 70]. To elicit events of communication breakdowns between PD and MAN that had led to defects related to the design and development, and manufacturing of software-intensive systems in concluded vehicle programs, some key informants at Volvo Cars were asked to recall and describe such communication events. In total, seven key informants were interviewed during 2012, and the selection of them was made primarily by expert judgment by representatives for PD and MAN at Volvo Cars. The selected key informants constituted a considerable part of the expertise at Volvo Cars and Table 6 shows the main characteristics of them. All of them had more than six years' experience of cross-functional work between MAN and PD in development of software-intensive automotive systems. Eliciting, analyzing, and verifying manufacturing and design requirements are central tasks of the key informants. They are also responsible for developing and maintaining the processes and tools managing the software-intensive systems in manufacturing. However, a balanced distribution was difficult to achieve at Volvo Cars, since staff in PD was unavailable for this study. This validity threat is further discussed in Sec. 6.3.

Table 6. Characteristics of the key informants.

Characteristics of the key informants	Number of key informants
Organizational affiliation	MAN (7) PD (0)
Roles	Manufacturing Engineers (4), System Manufacturing Engineer (1), Technical Expert (1), Line Manager (1)
Time at the company	Less than 1 year (0); 1–5 years (); 6–10 years (0); >10 years (7)
Experience of automotive software development	Less than 1 year (0); 1–5 years (0); 6–10 years (3); >10 years (4)
Gender	Male (7), Female (0)

When interviewing the informants, we used a focused interview strategy [63]. To collect relevant meta-data for sufficient descriptions and analysis of the identified communication events, an interview instrument was developed, including the model elements and their attributes in SCREAM, and some properties of interest for collecting and classifying supplementary information. When analyzing data, the properties also served as a coding scheme, helping to structure the data and model the communication events as CPs. Before the interviews, the research team (three researchers and one industry representative) brainstormed the properties mainly through discussing and analyzing different real communication events. The properties were built up, and the meaning of them was preliminary agreed upon. While collecting and analyzing the data, the research team continuously discussed and refined the properties (e.g. adding, splitting and reformulating them).

We considered some of the properties being generic for organizational communication, while others being more specific for the industrial setting investigated here. The properties are listed and exemplified in Tables 7 and 8, using one of the four

Table 7. Generic properties used in collection and classification of data for the communication events.

ID	Property	Description	Example
GP1	Initial reason for communication	Specifies the reason for initiating the communication. For example, changed information, task coordination, clarification and negotiation (e.g. requirements, design, tasks and decisions)	The manufacturing process has an impact on the maximum available time for calibrating the tailgate system. Therefore, requirements on calibrating the tailgate system (e.g. maximum available time for and demands on diagnostic services (e.g. ISO-14229–4 2012) for quality assuring and automating the calibration should be communicated from MAN (sender) to PD (receiver), and clarified and negotiated between them
GP2	Type of failing communication	Specifies the types of communication, which have failed and was central in the interaction between the actors. For example, formal communication (e.g. written and transferred specifications and structured meetings), and informal communication (e.g. unscheduled face-to-face meetings and email/phone interactions)	The manufacturing requirements on calibrating the tailgate system were formally specified and transferred to PD. But the information from the negation during the review meetings about how agreed solutions and implementations should actually look like was insufficient
GP3	Type of remedying communication	Specifies the types of communication, which was primarily used for remedying the defect	Improved calibration of the tailgate system was mainly discussed through unscheduled face-to-face meetings and email conversation combined with formal status reporting in a quality follow-up system
GP4	Implication cost for Actor 1 (MAN in this study)	Specifies the estimated implication costs for Actor 1 caused by the communication failure, including such as engineering costs and rises in product and manufacturing costs	Allocation of manufacturing resources for analyzing and creating solutions for calibrating the tailgate system had a cost of 7KUS$ and the defect caused an extra cost for opening and closing the tailgate manually in manufacturing of 0.35US$/vehicle over the life cycle of the car model. Implication cost = 7KUS$ + 0.35US$* (cars/year)*(car model life cycle)
GP5	Implication cost for Actor 2 (PD in this study)	Specifies the estimated implication costs for Actor 2 caused by the communication failure, including such as engineering costs and rises in product and manufacturing costs	Allocation of PD resources for analyzing and creating solutions for calibrating the tailgate system had a cost of 14KUS$. To resolve the defect once for all, the in-vehicle software was refactored for the next car model at cost of 28KUS$

Table 7. (*Continued*)

ID	Property	Description	Example
GP6	Too many iterations	Specifies whether the issue has been iterated several times between Actors 1 and 2, but no acceptable solution has been established	Although the calibration of the tailgate system has been iterated between MAN and PD several times, yet no acceptable solution has been implemented
GP7	Duplicate and ambiguous information	Specifies whether there are any external factors influencing or overruling the information exchange, agreements and decisions between Actors 1 and 2	PD and MAN have agreed on a solution for calibrating the tailgate system in manufacturing, but the demands on capitalizing on commonality between different vehicle models and brands overruled the agreement
GP8	Total estimated cost (GP4 + GP5)	Estimation of the total cost in US$ for the communication failure including implication costs for Actors 1 and 2 (GP4 + GP5)	When resolving the defect, MAN and PD allocated in total 21KUS$ (7 + 14) for analyzing and generating a feasible solution. The solution led to an extra cost of 28KUS$ for refactoring the software. Since it could not be implemented until the next car model, it caused an increased cost for calibrating the tailgate system in manufacturing of 0.35US$/vehicle over the life cycle of the current car model. This resulted in a total cost over the life cycle of the car model of 21KUS$ + 28KUS$ + 0.35US$* (cars/year)*(car model life cycle)

Table 8. Specific properties used in collection and classification of data for the communication events.

ID	Property	Description	Example
SP1	Defect detection point	Specifies when the defect was detected	Insufficient calibration of the tailgate system's automatic opening and closing function was detected in manufacturing during the trial production building phase at MS5.
SP2	Late detection of defect	Specifies whether the issues was discovered in late phases of PD. Late is defined as when defects are discovered after building and verifying prototypes of the complete vehicle at MS4	Yes, insufficient calibration of the tailgate system was first discovered in manufacturing at MS5
SP3	Defect originator direction	Specifies the direction from the actor causing the defect to the actor affected by the defect	The change of the automatic opening and closing function of the tailgate system was introduced by PD without considering the requirements on automation and quality assurance of the tailgate system in manufacturing, i.e. PD is the originator of the defect and the function change affects MAN
SP4	Defect delivered to customer	Specifies whether the defect can be delivered to the end-customers (users of the vehicle) and affect their experience	The manufacturing requirements on quality assurance of the calibration were not fulfilled, and thus deficient tailgate system functionality may be delivered to the customer
SP5	Introduction of new technology	Specifies new technologies introduced in vehicles or manufacturing processes causing the defect	A new technology for optimizing the automatic opening and closing of the tailgate has been introduced.
SP6	Variants	Specifies whether a certain combination of, or too many variants cause the defect	Insufficient calibration of the tailgate system is only identified for car variants with 5 doors
SP7	SW/HW related	Specifies whether the main cause of the defects can be referred to hardware or software, or both	The tailgate system calibration issue is primarily caused by inefficient calibration mechanisms in the in-vehicle software
SP8	Defect solution point	Specifies when and where a solution for solving the defect once for all, i.e. eliminating any further implication costs for the actors, was implemented and verified	The solution for improved calibration of the tailgate system functionality was implemented by PD and verified during building of prototypes of the next car model at MS4
SP9	Solution cost for solving the defect	Specifies the estimated costs for solving the defect once for all, i.e. eliminating any further implication costs for the actors	In order to adapt the tailgate system functionality so it can be calibrated in the manufacturing process without implication costs, refactoring of the in-vehicle software was implemented in the next car model at a cost of 28KUS$

examples of events. The generic properties are tagged GP1 and GP2, etc. and the specific ones are tagged SP1 and SP2, etc. For example, in this inquiry, the generic properties GP4 (implication cost for Actor 1) and GP5 (implication cost for Actor 2) include all the implication costs for MAN and PD, respectively. Such costs are, for example, allocated engineering resources for resolving the defect, and rises in product and manufacturing costs caused by the defect. GP4 and GP5 can be applied to other communication settings involving other actors as well (e.g. product managers and design engineers), and thus we consider them as generic. SP9 (solution cost for resolving the defect), on the other hand, is a specific property, since it is explicitly attributed to the costs for resolving defects once for all, so any further implication costs for MAN and PD are eliminated. SP9 is added to the implication costs but can be discerned as a separate cost when analyzing the meta-data.

All the interviews were held in Swedish with one interviewee and one interviewer, who was responsible for the interview process and took extensive notes organized according to the attributes and properties of the instrument. The interview time varied between 30 min and 60 min, including follow-up interviews to clarify and expand the descriptions during the analytic process. To enrich the understanding and analysis of the identified communication events, data (e.g. quantitative data on implication costs) were also extensively collected from pertinent documentation and archival records (e.g. process descriptions, project follow-up data and specifications). This data was also used to corroborate interview data.

4.2.2. *Step 2—Data modeling and classification*

The purpose of Step 2 was to achieve a consistent and accurate coding of interview data and supplementing information in documents based on the elements and attributes in the communication model and the properties of interest.

The procedure for modeling each of the communication events started with building and structuring the extracted data based on the elements and their attributes. This in order to obtain consistent syntax and semantics for all the events' descriptions of the communication between PD and MAN. Based on these descriptions the attributes were coded, and the elements chained, resulting in CPs representing the modeled characteristics of each communication event.

The data were also classified based on the properties listed in Tables 7 and 8. For example, estimated engineering costs for resolving the defect (e.g. data collection, analyzing, validating, reporting, and decision-making), and any increases in product and manufacturing cost (e.g. extra process costs for configuring, calibrating and verifying cars in manufacturing) were attributed to GP4 and GP5. If there was a once-for-all-cost that solved the defect, it was added to the implication costs for MAN (GP4) and PD (GP5) as a separate cost item. For example, if the defect was resolved by refactoring and releasing the in-vehicle software, this cost was attributed to SP9 (see Table 8) and added as a separate cost item in the implication cost for PD (GP5, see Table 7).

A preliminary modeling and classification of data were performed by one researcher and one industry representative in a series of meetings. The extracted data of each communication event was filled out and documented in an MS Excel sheet.

Furthermore, a short rationale, explaining the modeling and why the data should be classified into a specific property was added (e.g. why the event included a solution cost classified into SP9).

The results were then presented and discussed with two other researchers until an agreement was reached, so-called peer debriefing [64]. Mainly based on the accessibility and reliability of data, communication events were also selected for further data analysis. This resulted in a set of 16 events. Furthermore, while carrying out Step 2, the meaning of attributes and properties, and the orthogonality between them were continuously assessed and evaluated among the members of the research team to enhance the consistency and accuracy of the data being modeled and classified.

4.2.3. *Step 3—Data analysis*

In Step 3, the data were analyzed through content analysis [64]. The document analysis was guided by the results extracted from the interviews to enrich the analysis and understanding. Furthermore, for each of the events identified, relevant documentation was examined to check whether it indeed confirmed the interview data. If the documentation contained what was included in an event, it was considered to support it. Similarly, if the documentation contradicted what was included, it was considered not to support the event.

Step 3 included two main parts: (1) quantitative analysis of modeled and classified data, and (2) in-depth qualitative analysis of CPs and extracted meta-data. The first activity consisted of calculating the frequencies and costs of events for different CPs and codings of the single elements.

In the second activity, we used qualitative techniques for scrutinizing the modeled and classified data and confirming the results, such as triangulation of quotes of key informants, e-mail conversations, and documented information (e.g. specifications, processes, project follow-up data). Example quote: "In project x, PD did not understand how the manufacturing requirements would affect system x when they wanted to configure system x in the factory". This quote indicates that there are problems in communicating manufacturing requirements so they are understood, and was further analyzed by reviewing requirements specifications and descriptions of processes for balancing requirements across PD and MAN at Volvo Cars. To enhance the validity, similar quotes of other events and of other key informants and documented information were triangulated.

Finally, a root cause analysis of the results of the content analysis was performed with an overall aim to reveal underlying main causes and effects of the communication failures, setting the baseline for the creation of candidate solutions. For example, the examination of project follow up data showed that the manufacturing requirements were insufficiently broken down and balanced on the required design

level, ensuring the configuration of system x in manufacturing. Both the content analysis and the root cause analysis were performed by one researcher and one industry representative in a series of meetings. Furthermore, the results and the use of the model were shared and discussed with other researchers and industry representatives for PD and MAN at Volvo Cars, see Sec. 6.3.

5. Results and Analysis

Overall, 13 types of CPs were found among the 16 analyzed events and a grand total estimated improvement potential of 11 224.6KUS$. This indicates that the events provided a diverse sample of weak communication situations between PD and MAN. However, the identified CPs only represent 2.5% (13 out of 512) of all theoretically possible patterns that our model can express.

Table 9 shows the CPs found in the analyzed communication events and the number of such events linked to each of the CPs (there were four events of type CP01, the rest of the events found were each unique).

In the table, each CP is generically described, and the summarized estimated total cost (GP8, defined in Table 7) of the linked communication events for each of the CPs are given. For example, for the four communication events that are linked to CP01 the summarized estimated total cost (GP8) is 9 697.3KUS$, while only one, low-impact event is linked to CP13, showing a cost of 12.2KUS$. The CPs are presented in descending order according to the summarized estimated total cost.

To clarify the coding of the CPs in Table 9, we exemplify four different CPs, CP01, CP07, CP08, and CP11 in Table 10, using the four examples of events. Looking at CP01, in all of the events for this pattern, MAN act as the sender and PD as the receiver. For example, in one of the linked events, MAN has initiated the communication by sending manufacturing requirements on calibrating the tailgate system to PD (e.g. maximum time for the calibration and demands on diagnostic services (e.g. ISO-14229–4 2012) for quality assuring and automating the calibration. However, further information has not been exchanged, and the gradual detailing of requirements and solutions for calibrating the tailgate system should have, but has not, been specified. MAN has neither understood the impact of the requirements on the calibration of the tailgate system nor acted on the matter by adapting the affected manufacturing operations. PD has acted by changing the design prerequisites for the tailgate system but without considering the impact of the manufacturing requirements on the calibration of the tailgate system.

The high estimated total cost of 9697.3 KUS$ for CP01 is mainly related to the high impact of increasing the variable costs for each car, for example, added process time for downloading software to each car in manufacturing.

Table 8 also shows that there is a large difference in cost between, for example, CP13 and CP01 (9697.3 KUS$ and 12.2 KUS$, respectively). The explanation is that for CP13, in contrast to CP01, the costs merely includes costs for dedicated

Table 9. Identified CPs.

	Element/Attributes									
	Sender		Communication		Specification		Receiver			
ID	Understood	Acted on	Should be comm.	Has been comm.	Should be spec.	Has been spec.	Understood	Acted on	#events	Sum event cost (KUS$)[a]
CP01	MAN(nu)	MAN(na)	C(sc)	C(*c*)	S(ss)	S(ns)	PD(nu)	PD(*a*)	4	9697.3
CP02	MAN(*u*)	MAN(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	PD(*u*)	PD(na)	1	592.1
CP03	MAN(nu)	MAN(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	PD(nu)	PD(na)	1	358.4
CP04	PD(*u*)	PD(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	MAN(nu)	MAN(*a*)	1	136.6
CP05	PD(*u*)	PD(*a*)	C(sc)	C(*c*)	S(ss)	S(*s*)	MAN(nu)	MAN(na)	1	86.9
CP06	PD(nu)	PD(*a*)	C(sc)	C(*c*)	S(ss)	S(*s*)	MAN(nu)	MAN(na)	1	80.8
CP07	MAN(*u*)	MAN(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	PD(*u*)	PD(*a*)	1	65.6
CP08	PD(nu)	PD(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	MAN(nu)	MAN(*a*)	1	53.1
CP09	MAN(*u*)	MAN(*a*)	C(sc)	C(*c*)	S(ss)	S(*s*)	PD(nu)	PD(*a*)	1	37.9
CP10	PD(nu)	PD(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	MAN(*u*)	MAN(*a*)	1	37.9
CP11	PD(nu)	PD(*a*)	C(nsc)	C(*c*)	S(nss)	S(ns)	MAN(*u*)	MAN(*a*)	1	32.9
CP12	PD(nu)	PD(*a*)	C(sc)	C(nc)	S(ss)	S(ns)	MAN(nu)	MAN(na)	1	32.9
CP13	PD(nu)	PD(*a*)	C(sc)	C(*c*)	S(ss)	S(ns)	MAN(nu)	MAN(na)	1	12.2
Grand total									16	11 224.6

[a]The calculation of the estimated total cost for each event is based on 500.000 cars/year, 225 working days/year, 7 years product life cycle.

Table 10. Examples of CPs.

Element	CP01	CP07	CP08	CP11
Sender	MAN is the sender since MAN initiated the communication by sending manufacturing requirements on calibrating the end positions of the tailgate. MAN has **not acted** by adapting the manufacturing operations and has **not understood** the implications of the calibration.	MAN is the sender since MAN initiated the communication by sending manufacturing requirements on configuring the DDS (e.g. maximum available time for downloading software to the DDS ECU). MAN has **acted** by adapting the affected manufacturing operations as MAN has **understood** the impact of increasing the line speed on the maximum available time for downloading software to the DDS ECU.	PD is the sender since PD initiated the communication, by sending requirements and test cases to MAN for testing functions of the audio system. PD has **acted** by changing the audio system design but has **not understood** the impact of the manufacturing environment and processes on the test.	PD is the sender since PD has initiated the communication by sending requirements on verifying and documenting part and serial numbers for an infotainment component. PD has **acted** by changing the design so required data can be retrieved from the component but has **not understood** the necessity of doing it in manufacturing.
Communication	The manufacturing requirements on calibrations in the factory affect the software design of the calibration algorithm and quality assurance mechanisms, and **should be communicated** to PD and have been **communicated**.	Increasing the manufacturing line speed **should be communicated** to PD since it has an impact on the maximum available time for downloading software to the DDS ECU. MAN has **communicated** manufacturing requirements on the maximum available time for downloading software to the DDS ECU	Information about the test **should be communicated** to MAN and has been **communicated** to enable correct implementation of the audio function tests in manufacturing	Requirements on verifying an documenting part and serial numbers of the infotainment component were irrelevant and **should not be communicated** to MAN but has been **communicated**, i.e. redundancy

Table 10. (*Continued*)

Element	CP01	CP07	CP08	CP11
Specification	Requirements and solution for the tailgate calibration has been elaborated and **should be specified** but have **not** been **specified**.	Exchanged information for clarifying trade-offs between requirements and alternative solutions for handling the increased manufacturing line speed **should be specified** but has **not** been **specified**.	Information has been exchanged and developed test procedures and evaluation criteria **should be specified** but have **not** been **specified**.	Requirements and solutions for verifying and documenting part and serial numbers are irrelevant and **should not be specified** but has been **specified**.
Receiver	PD has **acted** by adapting the design prerequisites for calibrating the tailgate but has **not understood** the manufacturing requirements.	PD has **acted** by changing the design prerequisites for downloading software to the ECU has **understood** the impact of increasing the manufacturing line speed on the maximum available time for downloading software to the ECU.	MAN has **acted** by implementing new technologies for testing the audio system but has **not understood** implications of the design prerequisites for the audio system.	MAN has **acted** by implementing required changes of the manufacturing operations and has **understood** the requirements on verifying and documenting part and serial numbers of the component.

engineering resources to such activities as data collection, analyzing, reporting and decision-making, and no increased variable costs for each car.

In Secs. 5.1–5.3, we further analyze the results in regard to: (1) each single model element, (2) CPs, and (3) properties. The property analysis mainly includes the cost-related properties, namely GP4, GP5 and GP8 (see Table 7). We have chosen to focus on these properties because they show the potential benefits of improving the communication between PD and MAN at Volvo Cars, providing and important rational on which the need of further work on solutions to the communication problems can be motivated at Volvo Cars. Also, the other properties will provide important information in the search for solutions.

5.1. *Single element analysis*

The distribution of the communication events over the model elements and the codings of their attributes are shown in Fig. 3. The summarized estimated total cost (GP8) of the linked communication events for each coding of the element attributes is also given. For example, for the communication element, the summarized estimated total cost of the 14 communication events with the coding C (sc,c) is 11,158.8KUS$.

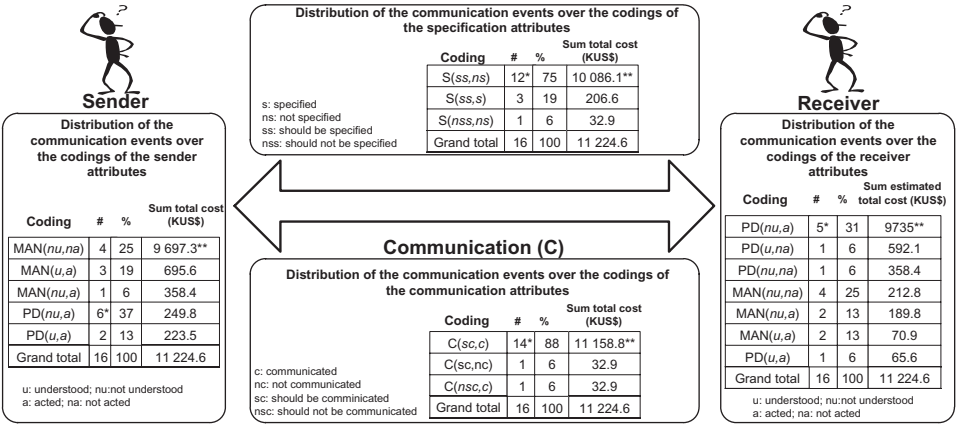

Fig. 3. Distribution of the events and their summarized cost over the codings of the attributes (*Most frequent, **Highest summarized estimated cost).

Figure 3 shows that most of the events are coded as C(sc,c) for the communication attributes (14 out of 16) while only two are coded as C(nsc,c) and C(sc,nc). This reflects the difficulties in identifying events where no communication has occurred, or the matter should not be communicated. Thus, it is unlikely to find events where the matter should not and has not been communicated, i.e. being coded as C(nsc,nc).

Looking at the sender attributes, 62.5% (five out of eight) of all possible codings are covered by the events. The most common codings of the sender attributes are

PD(nu,*a*) (37%) followed by MAN(nu,na) (25%). The codings MAN(nu,na) and MAN(u,*a*) have the highest summarized estimated total cost (9697.3KUS$ and 695.6KUS$, respectively).

There are also no events coded as PD(nu,na) and PD(*u*,na) where PD has not made any changes affecting MAN. One explanation for this may be a biased selection of the communication events. The 16 events were collected by informants from MAN, who consciously or unconsciously might tend to select events where PD had acted on the matter by changing the design, so it had an effect on the manufacturing operations. Another explanation is that PD primarily drives technology innovations, and in particular for software-intensive systems, which in turn demand adaptations of affected manufacturing operations [10].

In regard to the receiver attributes, Fig. 3 reveals that 88% (seven out of eight) of all possible codings are covered by the events. The high representativeness indicates a good spread of the sample of events. However, there are no events for MAN(*u*,na), which may also be a result of biased selection and analysis of the communication events. That is, when MAN has understood, there could have been a tendency to select events where MAN always has a willingness of taking required actions to solve the problem. Moreover, PD(nu,*a*) and PD(*u*,na) have the highest summarized cost (9735KUS$ and 592.1KUS$, respectively).

Looking at the distribution of events over the codings of the specification attributes in Fig. 3, a majority of the events (12 out of 16) shows that the exchanged information has been specified insufficiently, and the summarized total estimated cost of these events is 10,086.1KUS$. Moreover, in three of the events, the communication has failed even though the exchanged information had been sufficiently specified.

To get an overview of the representativeness of the different codings of the sender and receiver attributes, Fig. 4 displays the number of events for each combination of the attributes and the involved actors (PD and MAN), using a bubble graph [71]. The size of the bubbles indicates the number of events for each combination. For example, four events are linked to the combination where MAN is acting as the sender with the coding MAN(nu,na) (has not understood and not acted on the matter) and PD is acting as the receiver with the coding PD(nu,*a*) (has not understood, but has acted on the matter).

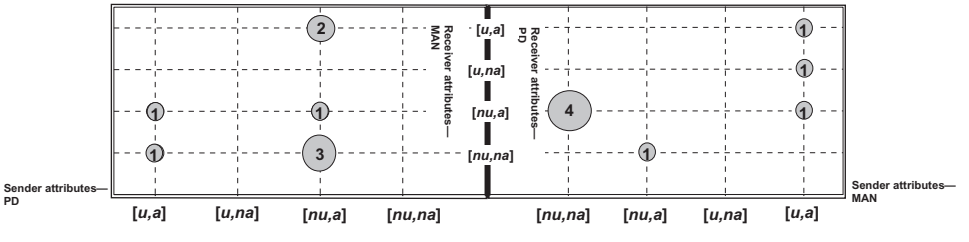

Fig. 4. Distribution of events over PD and MAN, and the codings of the sender and receiver attributes.

The graph shows also that in 14 of the 16 events, the communication has failed due to insufficiently shared understanding of the matter of concern. In addition, in nine of those 14 events, neither PD nor MAN had understood the matter, whereas either PD or MAN had understood in five events.

On the other hand, in the upper right quadrant of the graph, it can be observed that the communication has failed in two of the events, although both MAN (sender) and PD (receiver) have understood the matter. Moreover, there are no events being coded with the combination "na" for both the sender and the receiver (e.g. MAN(*u*,na) and PD(nu,na)). Thus, identifying events where neither the sender nor the receiver has taken any actions, affecting one or the other of them, seems unlikely.

5.2. *CP analysis*

Looking at the distribution of events over the CPs in Table 8, CP01 has the largest number of links (4) and the highest summarized estimated costs (9697.3KUS$). An analysis of the four events linked to CP01 shows that they have large deviations in regards to the cost, ranging from 37.9KUS$ to 6,199.8KUS$. This indicates that it is not possible to assume that events modeled as CP01 are usually the most costly in the setting investigated here.

For CP07 it can be seen that the communication has failed although both MAN and PD have understood and acted. The requirements and the alternative solution to resolve the too long time for downloading software to the DDS ECU for configuring the DDS had been understood and selected. PD has reduced the software size by modularizing configuration part of the software and MAN has adapted the tools for the downloading process (e.g. updating the software download equipment for handling new software download procedures and diagnostic services (e.g. ISO-14229–4:2012)) so only the configuration part of the software could be downloaded. The coding of CP07 indicates that an underlying cause of the communication failure can be related to specifications, since the exchanged information should have, but has not, been specified sufficiently. Looking into data, specifications on software parts pre-loaded at supplier and necessary adaptions in manufacturing for downloading the configuration software were missing.

CP11 shows that the communication has been established between PD and MAN even though it should not. This because PD had not understood the necessity of reading the part and serial number of the infotainment component to ensure that vehicles contain the right configuration in accordance with corresponding product specifications and to document each vehicle's configuration by recording required data in a vehicle database before the vehicle leaves manufacturing. This led to an estimated total cost of 32.9KUS$ for unnecessary work (refactoring of the software enabling reading of part and serial number from the infotainment component and implementation of required adaptations in the manufacturing process), which can be referred to redundancy and waste of resources. Moreover, for CP11 and CP10, the

matter has been understood and acted on by MAN despite the fact that PD has not understood and the exchanged information has not been specified sufficiently.

As mentioned in Sec. 5.1, lack of shared understanding is prevalent in the communication events examined. This is also reflected in Table 9 where either PD or MAN, or neither of them, have understood the matter of concern in 11 out the 13 CPs. The summarized estimated cost for these CPs is 10 566.9KUS$. In five of those 11 CPs (CP04, CP05, CP09, CP10 and CP11), either PD or MAN have understood, but the sharing and building of knowledge between them have failed. One reason for this may be a lack of specifications, since three out of these five CPs show that the matter has not been specified.

In contrast, the other six CPs (CP01, CP03, CP06, CP08, CP12 and CP13) show that neither of PD or MAN has understood the matter. Of these, the matter was not specified in five CPs. However, requisite knowledge is not available within the two departments, and thus the communication problems for these CPs are most likely stemming from other factors than insufficient sharing of knowledge, and inadequate specifications.

To gain a better understanding and find out the effects and possible causes for the communication failures of the different CPs (e.g. the high cost for CP01 and why the apparently flawless communication failed in CP07), the following presents a drill-down analysis of the linked events by using collected meta-data.

5.3. *Property analysis*

The summarized estimated total costs for the CPs, GP8 (see Table 7), were calculated to 11,224.6KUS$, see Table 9. The distribution of the implication costs for PD and MAN (see GP4 and GP5 Table 7) over the CPs showed that GP4 constitutes 96% (10,804.7 of 11,224.6) of the grand total estimated cost for all the CPs. This indicates that the largest improvement potential for the set of communication events investigated here is related to implications for manufacturing (e.g. increased process time and unstable processes causing unnecessary reprocessing of the cars). In the tailgate calibration event, for example, the manufacturing requirements on maximum calibration time and automatization was not fulfilled. This caused extra station in the manufacturing process for manually closing and opening the tailgate, increasing the running costs. We also discovered that in 14 of the 16 events the defects was detected late (see SP2 in Table 8) and that solutions for resolving the defect and eliminating the implication costs once and for all had not been developed and agreed upon, and implemented in seven of the 16 events. Furthermore, the estimated solution cost (SP9 see Table 8) was relatively low compared to the implication costs. For example, the solution cost for refactoring the in-vehicle software to resolve the issue in the tailgate calibration event linked to CP01 could roughly be estimated to 28KUS$ (SP9), since it could not be implemented in the car model produced the estimated implication cost for MAN and PD (GP4 and GP5) is 9697.3KUS$ over its life cycle.

With regard to CP07, the property GP7 (duplicate and ambiguous information), see Table 7, showed that mutual understanding and an initial agreement on a solution for configuring the DDS were obtained between PD and MAN, but not communicated to a third party involved in the development (here another brand). Furthermore, the completeness of the specifications was not attained, since demands on common technical solutions between the brands were not included. This led to that the agreed solution was first overruled, but later developed and implemented owing to a too high risk of quality problems and low efficiency in manufacturing. A more thorough specification of the prior agreed solution between PD and MAN and inclusion of the commonality demands might have been helpful in the argumentation for the agreed solution in the discussions with the third party.

In the analysis, we also looked for the main causes for the communication failure. The single element analysis showed that in 12 out of the 16 events the matters of concerns were insufficiently specified. The analysis of the meta-data for these events revealed that six of these 12 events are concerned with problems of communicating manufacturing requirements on the software-intensive systems. Furthermore, the summarized estimated cost for these events constitutes 92% of the total estimated cost (10,355 of 11,224.6 KUS$).

Upon further looking into the meta-data, primarily GP1, GP2 and SP1 (see Tables 7 and 8), and extracted background information of these events, we found that MAN had initiated the communication in early phases of the car development cycle (before MS2) through specifying and communicating generic and high-level (complete vehicle level) manufacturing requirements (e.g. the duration time of calibration shall not exceed 10 s), into Volvo Cars' requirements management system. However, after PD's acceptance of the requirements at MS2, no further balancing of the manufacturing and product requirements, and detailing of the specifications, providing information about how the actual implementations should look like in the manufacturing processes on systems and component level, were performed. For example, the solution for achieving the maximum calibration time of 10 s by dividing the tailgate calibration into two algorithms and how to execute them at two different stations in the manufacturing process were not specified. We also found a lack of formal methods and practices, supporting the breakdown of the requirements on lower levels and further balancing of them across MAN and PD in later stages of the car development cycle (MS2-MS6). For example, the manufacturing requirement on maximum calibration time was not broken down to the sub-system for maneuvering the tailgate and the software component controlling the calibration and logics of the automatic open-and-close of the tailgate.

Based on these findings, Volvo Cars expressed a direct need of improving the processes for balancing requirements and specifying the exchanged information between PD and MAN after MS2. Given the CPs generated by using SCREAM and the improvement potential of 10,355KUS$, a follow-up study was therefore conducted in close collaboration with Volvo Cars. This study is reported in [67] and resulted in a lightweight RE framework focusing on improving requirements communication and

specifications, and initial feedback from using the framework yielded promising results.

6. Discussion

Section 6.1 discusses the findings related to the specific results of the Volvo Cars case presented in Sec. 5, and in Sec. 6.2 our findings related to the use of SCREAM in practice. Section 6.3 evaluates the validity threats to the study presented in this paper.

6.1. *Findings related to Volvo cars case*

The analysis of the CPs showed that in five of all the CPs either PD or MAN had adequate knowledge about the matter of concern. In three of these five CPs, the matter had been insufficiently specified which can be an explanation for the difficulties in sharing and building knowledge between PD and MAN. This is in line with Almefelt *et al.* [11], who found that particularly manufacturing knowledge is often experience-based and tacit rather than being captured in detailed specifications of purposes and goals. For example, manufacturing requirements on configuring the DDS ECU in manufacturing are generic and on the complete vehicle level, making it difficult for developers of a single function system, or component to convert the constraints to measurable and understandable parameters. Difficulties in specifying and communicating precise and understandable requirements on an appropriate level of abstraction are also well-known problems in software development [72, 73]. Furthermore, Calefato *et al.* [28] found that text-based communication was preferred in RE because of its ability to provide sufficient documentation and make decisions clear.

One strategy for building knowledge in an organization is systematization and storage of explicit (codified) knowledge gained from concluded projects and making it accessible and easy to use by anyone at the company. In lean automotive companies, this is commonly accomplished by establishing know-how databases evolved from checklists and A3s, see Morgan and Liker [6], and similarly, this is often referred to as having an Experience Factory (EF) [74] in software development. Examples of benefits are improved quality of produced development artifacts, more effective risk management throughout the development process, and reduced risk of propagating the same mistakes across projects [75]. However, this strategy is often resource consuming [74], and the need for communication is probably not reduced by increasing the documentation [15]. Therefore, to build organizational knowledge other strategies, such as the creation of learning networks that encourage and facilitate the informal transfer of tacit knowledge throughout the company must also be adopted [6]. Also, agile software development approaches highly rely on the organization's capability of mediating tacit knowledge [76].

The modeling of the events also resulted in six CPs where neither MAN nor PD had understood the matter communicated. Improving the sharing of knowledge

between PD and MAN for these CPs seems therefore not to be enough. Furthermore, even if other measures are taken that directly can improve the communication itself (e.g. adding review meetings, communication media, and development artifacts), there are most likely other indirect factors that also have to be considered. For example, lack of fit between structural levels due to organizational differences between PD and MAN, and factors hindering knowledge transfer because of inadequate training of the staff, documentation (e.g. manufacturing and development processes, and requirements specifications) and ineffective IT-based tools [8, 50]. Uncertainties regarding roles and responsibilities within and across departments and project teams may also be a critical factor [8, 9]. More specifically, Pernstål *et al.* [10] reported that the responsibilities between MAN and PD that ensure a fit between the design of the software-intensive systems and the manufacturing processes are unclear and not always understood. Furthermore, in earlier work, low understanding of each other's roles has been found to be a critical factor in the PD/MAN interface for a successful production start, but also a major cause for gaps in software requirements communication [8, 19].

In the property analysis, we found that the solution costs (SP9) were relatively low compared with the implication costs (GP4 and GP5, see Table 7). This indicates that, rather than negotiating and establishing solutions resolving the defect once and for all, implication costs are allowed, especially when it comes to manufacturing. Since in 14 of the 16 events the defects were detected late (see SP2 in Table 8) one explanation for this can be the late discovery of the defects causing an unwillingness of introducing product changes due to a risk of jeopardizing the start of production.

A specific finding of this study is that the specification quality of upfront communicated high-level manufacturing requirements is not good enough. Improving the quality of the requirements specifications will probably, however, have a minor effect, since "perfect" requirements specifications are impossible to achieve. In particular, when it comes to writing textual requirements on large complex systems (e.g. [73, 77]). Furthermore, involved people and roles have a tendency to assume that upfront produced artifacts convey all the information needed for downstream development work, inhibiting the continuous exchange of information throughout the car development cycle, and in particular, across organizational boundaries [16]. Looking further into the handling of the manufacturing requirements revealed a lack of mechanisms supporting breakdown and both formal and informal communication of them throughout the full car development cycle. Furthermore, handover points where requirements information are conveyed between different roles, both within PD and MAN, and across them, were found. Handover points are critical for continuity of requirements communication, causing deteriorated awareness of the requirements and increased risk of ignoring them after the handover [19].

Referring to the goal-oriented requirements communication model, as described in Fricker *et al.* [27], the degree of a bidirectional communication of the manufacturing requirements seems to be much like the informationless paradigm, i.e. MAN formulates the requirements and transfers them over to PD who receives and accepts

them, but without any further information exchange. Although techniques such as handshaking of requirements with improvement proposals, as described in Fricker *et al.* [61], must be tailored so it fits the setting investigated here (e.g. dynamic roles of goal seeker and implementer, and the content of exchange information), its basic constituents can serve as inspiration and catalyst for creating pertinent solutions. It is also recommended that Volvo Cars looks into possibilities to improve the handovers (e.g. improving organizational and spatial proximities). For example, improving the passing of manufacturing requirements from manufacturing engineers working in early development phases (MS1–MS2) to those working in later phases (MS2–MS6), see Pernstål *et al.* [67].

6.2. *Findings related to the use of SCREAM*

One major finding is the capability of SCREAM to extend the analysis of the communication events by using the attributes b2 and b8 (has acted on) and the properties in Tables 7 and 8. The contribution theory state that successful communication is achieved when the actors has reached a common understanding, i.e. State 3, where pertinent documentation of exchanged information is particularly critical during the grounding process [39]. Basing on this, successful communications should be coded as sender(u);receiver(u);C(sc,c);S(ss,s). In addition, SCREAM models if actions has been taken or not taken by the sender and receiver related to the matter communicated. To be able to take the correct actions, a basic assumption is that both the sender and the receiver have understood the matter. This means that flawless communication taking the actions of the actors affecting them into consideration are most likely coded as sender(u,a);receiver(u,a);C(sc,c);S(ss,s). For example, even though both PD and MAN have understood the matter in CP02, the communication event was a failure since PD has not taken any actions primary due to late discovery of the defect and risk of jeopardizing start of production (see SP2 in Table 7). On the other hand, despite that both PD and MAN have understood and acted on the matter in CP07 the communication failed, since the property GP02 showed that the matter had not been communicated to a third party. Furthermore, reaching a common understanding might as well lead to that the correct action is no action taken. Consequently, even though CPs coded as both the sender and receiver has understood and the exchanged information has been specified sufficiently indicate that the communication itself has worked, the whole communication event is successful or unsuccessful depending on whether correct actions have been taken by the actors.

Collecting and analyzing data to ensure correct coding of the attributes has been understood (b1 and b7) and has been specified (b6) were not an easy task, and in particular, measuring the level of understanding [39]. To determine if the matter has been understood and the completeness of the specifications, we performed content and in depth analysis based on the data classified into the properties listed in Tables 7 and 8. In CP01, the gaps within the grounding process confirming that PD

and MAN had not reached a common understanding was extracted by especially examining the properties GP1, GP2 and GP3. GP1 and GP2 provided information about how and what was discussed and specified during the negotiation of the manufacturing requirements and from GP3 information communicated about solutions for remedying the defect were extracted. For example, one of the key informants said "during the negotiation we talked about the manufacturing requirements and the new tailgate system but we never discussed and analyzed how solutions for the calibration actually should work in the plant". Systems and software specifications, such as system requirement descriptions (SRD) and software requirement specifications (SWRS), confirmed this quote since documentation of solutions and system adaptations were lacking. In GP3, such as email conversations and notes of the quality follow-up system gave the possible to see how the solutions for resolving the tailgate calibration were discussed and adapted step by step to fit PD and MAN needs, and how the agreed solution was specified in, for example, the SRD and SWRS.

A major challenge was to attain orthogonality between, and clarity and precision of the model elements and their attributes. In particular, attaining non-overlap between the attributes b4 (has been communicated), and b2 and b8 (has acted on) was difficult. In order to discern b2 and b8 from b4, we defined b2 and b8 as any actions taken by an actor that have affected the other actor, except actions related to the communication itself (see also Sec. 3). However, even though the coding of these attributes was discussed and consolidated in revision meetings, their difference was sometimes perceived as subtle.

Establishing a unified understanding and interpretation of the attributes among the members of the review team is important for achieving accuracy and consistency of the modeling. In addition to our revision meetings, inter-rater agreement values (e.g. Fleiss Kappa) could have been computed for a pilot on a number of randomly selected events. However, limitations in allocating the necessary resources for identifying additional events, and collecting and compiling required data for the pilot made this unfeasible. Alternatively, the model could have been more fine-grained (e.g. including more and more precise model elements and attributes), but under the sacrifice of the scalability of the model.

We also found difficulties in avoiding interdependencies between the attributes. For example, the coding of one of the CPs (CP11) indicates that the communication was unnecessary since it should never have been established. Looking into the linked event, PD introduced an extended method for verifying and documenting the configuration of the cars, which showed out to be unnecessary. The information between PD and MAN was exchanged on an informal basis and although it was deemed irrelevant and should not have been specified, it could have been beneficial in this event. While specifying, PD could have had a better opportunity of utilizing MAN's understanding, and question the necessity of implementing the method. However, coding the specification attribute to "should be specified" while the communication attribute is coded to "should not be communicated" may be taken as illogical

reasoning and inconsistency. This points out difficulties in achieving full orthogonality between the communication and specification attributes owing to logical interdependencies between them.

For some events, there was a lack of relevant information, making it troublesome to identify the initial message and classify who act as the sender and the intended receiver. Using several key informants and asking them follow-up questions, and extensively looking for supplementing the information in documents and archival data were sometimes helpful, but we found many dead ends. However, this classification is not critical, since SCREAM is more concerned with interpreting and understanding the communication breakdown through analysis of the attributes of the elements. Furthermore, SCREAM does not assign the sender or the receiver roles such as goal seeker and implementer [27], since the distinctions between such roles are sometimes unclear and they may be shifted during the project cycle. For example, MAN can start as the problem owner, but when alternative solutions have been evaluated, the most beneficial alternative may be a change of the manufacturing processes, i.e. MAN has transformed from goal seeker to implementer.

A limitation of SCREAM is that it can only be used to model and assess communication events that involve two actors as there can be additional actors, for example, brokers, influencing the communication [22, 26]. This is reflected in CP07 where both MAN and PD had understood and acted on the matter, but the property analysis showed that the agreed solution for configuring the DDS was overruled by a third actor. However, including more actors in SCREAM will dramatically increase the complexity. Adding one actor can theoretically increase the possible codings of the CPs from 512 to 5123. Therefore, we instead decided to include this aspect of communication in the properties (see GP7 in Table 7). To further investigate the communication between the third actor and Actors 1 and 2, and at the same time obtain scalability in such complex communication situations, instantiations of SCREAM can be used (e.g. between PD at Volvo Cars and PD at the other brand). Moreover, the event shows that attaining the right understanding and acting are not always enough since external actors with diverging understanding and interests can influence the communication established between two actors.

The total effort (time spent) used by both the company staff and the research team members for collecting and analyzing, and validating the results for the 16 events could be estimated to roughly 200 man-hours. A major part was dedicated to collect information about each communication event. However, we believe that the effort would be higher for this without using SCREAM as its elements, attributes and properties helped to collect and classify relevant data in a systematic way. Furthermore, the feedback on SCREAM from other researchers and professionals indicated that knowledge and experience of the communication situation investigated among practitioners using SCREAM are more critical than building their skills in using the model.

6.3. *Threats to validity*

One major threat in this study is that biased data are used since data were mainly collected from interviews with key informants currently representing MAN. To alleviate this threat, we used one informant at MAN that had considerable experience of designing automotive software systems at PD. Furthermore, we used interview data from multiple key informants and additional data sources including pertinent documentation and archival records for corroborating and triangulating data. Observer triangulation and prolonged involvement were also utilized [64]. Two of the members of the research team (one researcher and one industry representative) could frequently and extensively discuss and analyze the collected data. Moreover, they had more than 10 years of experience of cross-functional work between PD and MAN in development of software-intensive automotive systems at Volvo Cars. These measures were used to guard against observer and respondent bias.

When applying SCREAM, another threat is the quality of the modeling and classification of collected data (Step 2 in Sec. 4.2.2) and the results of the analysis (Step 3 in Sec. 4.2.3). This threat was mitigated primarily by triangulating data sources (see above) and utilizing peer debriefing [64]. The research team iteratively shared and reviewed the coding of data and the results of the analysis in a series of revision meetings to enhance the consistency and accuracy of the results. Furthermore, to receive feedback on the results and SCREAM, meetings and seminars were held with other researchers and representatives for PD and MAN at Volvo Cars (e.g. line and project managers and software systems and manufacturing engineers) who were familiar with the study.

Another major threat is that the key informants may not express their real opinions in the meetings and thus do not contribute with their expertise. To address this threat, the key informants were guaranteed anonymity and that sensitive information would neither be published nor possible to trace to individuals. The impression was that all the key informants spoke freely and were actively involved in the meeting.

In this study, we have solely examined communication events between PD and MAN at one company, namely Volvo Cars, limiting the applicability of the findings beyond the company studied. To enhance the possibilities for the readers to judge whether SCREAM and the results can be used in other settings, and in particular, in the car industry, we provide a thorough description of the industrial setting at Volvo Cars in Sec. 4.1. Furthermore, the description shows that many of the characteristics of Volvo Cars are also common in the car industry, especially in the premium car segment. Moreover, to be in a leading position, premium carmakers must continuously develop and introduce new innovations and technologies in products and manufacturing, which are usually adopted by other carmakers. Since SCREAM is primary based on combining central and well-established elements of communication, we believe that it has the potential to help practitioners analyzing their

communication problems in other industrial software development settings (e.g. lean and agile contexts) as well as in hardware oriented development. Even though these considerations may mitigate the threats to external validity, SCREAM was developed and applied to a specific industrial setting, and thus further studies must be conducted in order to strengthen the possibility to generalize the findings.

To mitigate threats to the reliability of the study, a thorough description and exemplification of the application process of SCREAM at Volvo Cars is provided in Sec. 4.2.

7. Conclusion

In software development, coordination and communication within and across organizational boundaries throughout the software development cycle are acknowledged as critical factors for success, especially when software is developed on a large scale.

In this paper, we present SCREAM, a novel communication model, that help researchers and practitioners perform postmortem analysis of events where weak communication have taken place and caused problems. The overall goal of SCREAM is to be conceptually simple and practically useful in different industry contexts and characterize organizational communication problems in a structured and descriptive way to reveal effects and causes on which efforts for developing improvements can be motivated and based.

SCREAM evolved in close cooperation with industry and was evaluated by applying it to 16 communication events, focusing on inter-departmental communication between PD and MAN in the design and development of software-intensive automotive systems at the Swedish automotive company Volvo Cars. Unusually for empirical SE research, our research methodology allowed us to collect data on the implication costs of the communication problems. Through feedback and discussions with the practitioners, and based on the results from using SCREAM we conclude

- When examining the communication events, SCREAM provides a structured and systematic way for collecting, modeling and classifying data.
- Using the model allows identification and analysis of CPs representing central elements of communication on a high-level, but also a detailed analysis of individual communication elements of each event, which can help in revealing underlying causes and effects for communication failures.
- Applying SCREAM at Volvo Cars, showed that lack of shared understanding of the matter being communicated is prevalent in the communication events examined.
- The resulting CPs showed that in many of the analyzed events a more detailed specification, and more and/or better communication in order to collectively agree on that specification, would be needed.

- The total estimated cost for the analyzed communication events was 11.2MUS$ and a detailed analysis revealed that manufacturing and product requirements are insufficiently balanced and detailed over the full car development cycle, causing an estimated implication cost of 10,355KUS$ for managing late defects and changes.

Our study shows that SCREAM is a useful tool in analyzing organizational communication problems in software development. Overall, the results of using SCREAM show that it is feasible and applicable in an industrial setting. The experience indicates that using central elements of communication for analyzing and structuring whole series of messages on a high-level instead of in-depth analysis on individual message level have a positive effect on ease to use and usefulness, and resources. In review meetings, the company agreed on the main results using SCREAM and building on them, the company is continuing the improvement initiative. Consequently, a follow-up study was conducted in close collaboration with Volvo Cars where a flexible and lightweight RE framework was developed to resolve the company's communication problems identified in the process of balancing requirements and solutions and evaluated with promising results.

In this paper, SCREAM has been used in one specific industrial setting. However, communication is and will continue to be a key concern in most software development projects. We believe that our proposed model could have value in many contexts and intend to further evaluate the potential of the model to satisfy the needs of other industrial settings.

Acknowledgments

The authors would like to express their gratitude to all those involved in this study and especially the interviewees at Volvo Cars.

References

1. M. Broy, I. H. Kruger, A. Pretschner and C. Salzmann, Engineering automotive software, *Proc. IEEE* **95**(2) (2007) 356–373.
2. K. Venkatesh Prasad, K. M. Broy and I. Krueger, Scanning advances in aerospace & automobile software technology, *Proc. IEEE* **98** (2010) 510–514.
3. J. Nihtilä, R&D–Production integration in the early phases of new product development projects, *J. Eng. Technol. Management* **16**(1) (1999) 55–81.
4. M. E. Conway, How do committees invent? *Datamation* **14** (1968) 28–31.
5. D. L. Parnas, On the criteria to be used in decomposing systems into modules, *Commun. ACM* **15** (1972) 12.
6. J. M. Morgan and J. K. Liker, *The Toyota Product Development System: Integrating People, Process, and Technology* (Productivity Press, NY, 2006).
7. D. K. Sobek, A. C. Ward and J. K. Liker, Toyota's principles of set-based concurrent engineering, *Sloan Management Rev.* **40**(2) (1999) 67–83.
8. A. Vandevelde and R. Van Dierdonk, Managing the design-manufacturing interface, *Int. J. Operations Production Management* **23** (2003) 1326–1348.

9. N. Lakemond, G. Johansson, T. Magnusson and K. Safsten, Interfaces between technology development, product development and production: Critical factors and a conceptual model, *Int. J. Technol. Intelligence Planning* **3**(4) (2007) 317–330.
10. J. Pernstål, A. Magazinius and T. Gorschek, A study investigating challenges in the interface between product development and manufacturing in the development of software intensive automotive systems, *Int. J. Softw. Eng. Knowledge Eng.* **22** (2012) 965–1004.
11. L. Almefelt, F. Berglund, P. Nilsson and J. Malmqvist, Requirement management in practice: Findings from an empirical study in the auto-motive industry, *Res. Eng. Design* **17**(3) (2006) 113–134.
12. V. Sumantran, Accelerating product development in the automobile industry, *Int. J. Manufacturing Technology Management* **6**(3/4) (2004) 361–371.
13. A. Peters, E. Rooney, J. Rogerson, R. McQuater, M. Spring and B. Dale, New product design and development: A generic model, *TQM Magazine* **11**(3) (1999) 172–179.
14. J. R. Galbraith, *Designing Complex Organizations* (Addison-Wesley Longman, Boston, 1973).
15. B. Curtis, H. Krasner and N. Iscoe, A field study of the software design process for large systems, *Commun. ACM* **31** (1988) 11.
16. R. E. Kraut and L. A. Streeter, Coordination in software development, *Commun. ACM* **38** (1995) 69–81.
17. K. Schwaber and M. Beedle, *Agile Software Development with Scrum* (Prentice Hall, Upper Saddle River, 2001).
18. M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit* (Addison-Wesley, Boston, 2003).
19. E. Bjarnason, K. Wnuk and B. Regnell, Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development, in *Proc. 19th IEEE Int. Requirements Engineering Conf.* 2009, pp. 37–46.
20. M. Cataldo, M. Bass, J. D. Herbsleb and L. Bass, On coordination mechanisms in global software development, in *Proc. Int. Conf. Global Software Engineering*, 2007, pp. 71–80.
21. M. Cataldo, J. D. Herbsleb and K. M. Carley, Sociotechnical congruence: A Framework for assessing the impact of technical and work dependences on software development productivity, in *Proc. 2nd Int. Symposium on Empirical Software Engineering Measurement*, 2008, pp. 2–11.
22. S. Marczak, D. Damian, U. Stege and A. Schröter, Information brokers in requirement-dependency social networks, in *Proc. IEEE 16th Int. Conf. Requirements Engineering*, 2008, pp. 53–62.
23. S. Marczak, I. Kwan and D. Damian, Investigating collaboration driven by requirements in cross-functional software teams, in *Proc. Int. Workshop on 2009 Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills*, 2009, pp. 15–22.
24. I. Kwan and D. Damian, Extending socio-technical congruence with awareness relationships. in *Proc. 4th Int. Workshop on Social Software Engineering*, 2011, pp. 23–30.
25. I. Kwan, A. Schröter and D. Damian, A weighted congruence measure, in *Proc. Socio-Technical Congruence Workshop at ICSE Conf.*, 2009, pp. 1–10.
26. I. Kwan, A. Schröter and D. Damian, Does socio-technical congruence have an effect on software build success? A study of coordination in a software project, *IEEE Trans. Software Eng.* **37** (2011) 307–324.
27. S. Fricker, T. Gorschek and M. Glinz, Goal-oriented requirements communication in new product development, in *Proc. Second IEEE Int. Workshop on Software Product Management*, 2008, pp. 27–34.

28. F. Calefato, D. Damian and F. Lanubile, Computer-mediated communication to support distributed requirements elicitations and negotiations tasks, *Empirical Software Engineering* **17** (2012) 640–674.
29. R. T. Craig, Communication theory as a field, *Communication Theory* **9** (1999) 119–161.
30. K. Miller, *Communication Theories* (McGraw-Hill, 2002).
31. S. W. Littlejohn and K. A. Foss, *Theories of Human Communication* (Wadsworth Publishing, 2007).
32. J. Fiske, *Kommunikationsteorier, En Introduction* (Wahlström & Widstrand, Stockholm, Sweden, 1994).
33. S. Windahl and D. McQuail, *Kommunikationsmodeller* (Studentlitteratur, Lund, Sweden, 1978).
34. C. E. Shannon, A mathematical theory of communication, *Bell System Technical J.* **27** (1948) 379–423 and 623–656.
35. R. Jakobson, The speech event and the functions of language, in *On Language* (Harvard University Press, 1990), pp. 69–79.
36. L. Forsdale, *Perspectives on Communication* (Random House, 1981).
37. K. Giffin and B. R. Patton, *Fundamentals of Interpersonal Communication* (Harper & Row, 1971).
38. H. H. Clark and E. F. Schaefer, Contributing to discourse, *Cognitive Science* **13** (1989) 259–294.
39. H. H. Clark and S. E. Brennan, Grounding in communication, in *Perspectives on Socially Shared Cognition* (American Psychological Association, 1991), pp. 127–149.
40. S. Brown and K. Eisenhardt, Product development: Past research, present findings, and future directions, *Acad. Management Rev.* **20** (1995) 343–378.
41. M. Tushman and D. Nadler, Information processing as an integrating concept in organization design, *Acad. Management Rev.* **3**(3) (1978) 3613–3624.
42. J. Short, E. Williams and B. Christie, *The Social Psychology of Telecommunications* (Wiley, New York, 1976).
43. R. L. Daft and R. H. Lengel, *Information Richness: A New Approach to Managerial Behavior and Organization* (JAI Press, 1984).
44. R. L. Daft and R. H. Lengel, Organizational information requirements, media richness and structural design, *Management Sci.* **32** (1986) 554–571.
45. J. Fulk and B. Boyd, Emerging theories of communication in organizations, *J. Management* **17** (1991) 407–446.
46. R. S. Burt, Models of network structure, *Ann. Rev. Sociol.* **6** (1980) 79–141.
47. R. E. Rice and W. D. Richards Jr., An overview of network analysis methods and programs, in *Progress in Communication Sciences*, Vol. 6 (Ablex, 1985), pp. 105–165.
48. P. R. Monge and N. S. Contractor, Emergence of communication networks,. in *The New Handbook of Organizational Communication Advances in Theory*, 2000, pp. 440–502.
49. P. Hinds and S. Kiesler, Communication across boundaries: Work, structure, and use of communication technologies in a large organization, *Organization Sci.* **6** (1995) 373–393.
50. L. Argote, *Organizational Learning: Creating, Retaining and Transferring Knowledge* (Springer, 2013).
51. J. W. Meyer and B. Rowan, Institutional organizations: Formal structures as myth and ceremony, *Amer. J. Sociol.* **80** (1977) 340–363.
52. K. Ehrlich and M. Cataldo, All-for-one and one-for-all?: A multi-level analysis of communication patterns and individual performance in geographically distributed software development, in *Proc. ACM CSCW* (2012), pp. 945–954.

53. K. Ehrlich and M. Cataldo, The communication patterns of technical leaders: Impact on product development team performance, in *17th ACM Conf. Computer Supported Cooperative Work and Social Computing*, 2014, pp. 373–385.
54. H. Mintzberg, *The Structure of Organizations* (Prentice Hall, 1979).
55. F. P. Brooks, No silver bullet essence and accidents of software engineering, *Comput.* **20**(4) (1987) 10–19.
56. C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity* (MIT Press, 2000).
57. M. E. Sosa, S. D. Eppinger and C. M. Rowles, The misalignment of product architecture and organizational structure in complex product development, *Management Sci.* **50** (2004) 1674–1689.
58. M. Cataldo, P. A. Wagstrom, J. D. Herbsleb and K. M. Carley, Identification of coordination requirements: Implications for the design of collaboration and awareness tools, in *Proc. Conf. Computer-Supported Cooperative Work*, 2006, pp. 353–362.
59. K. Ehrlich, M. Helander, G. Valetto, S. Davies and C. Williams, An analysis of congruence gaps and their effect on distributed software development, in *Proc. Socio-Technical Congruence Workshop at ICSE Conf.*, 2008.
60. G. Klir, *Facets of Systems Science* (Kluwer Academic, 2001).
61. S. Fricker, T. Gorschek, C. Byman and A. Schmidle, Handshaking with implementation proposals: Negotiating requirements under-standing, *IEEE Softw.* **27** (2010) 72–80.
62. M. Ivarsson and T. Gorschek, A method for evaluating rigor and indus-trial relevance of technology evaluations, *Empirical Software Eng.* **16** (2011) 365–395.
63. R. Yin, *Case Study Research: Design and Methods*, 3rd edn. (Sage, 2003).
64. C. Robson, *Real World Research: A Resource for Social Scientists and Practitioners-Researchers*, 2nd edn. (Blackwell, 2002).
65. ABG. V-Model: *Development Standard for IT-Systems of the Federal Republic of Germany* (Lifecycle Process Model, 1994).
66. R. G. Cooper, *Winning at New Products*, 3rd edn. (Perseus, 2001).
67. J. Pernstål, T. Gorschek, R. Feldt and D. Floren, Requirements communication and balancing in large-scale software-intensive product development, *Inf. Software Technol.* **67** (2015) 44–64.
68. F. Charfi and F. Sellami, Overview on dependable embedded systems in modern automotive, in *Proc. IEEE Int. Conf. Industrial Technology*, 2004, pp. 781–786.
69. J. C. Flanagan, The critical incident technique, *Psychol. Bull.* **51** (1954) 327–359.
70. D. D. Gremler, The critical incident technique in service research, *J. Serv. Res.* **7** (2004) 65–89.
71. K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson, Systematic mapping studies in software engineering, in *Proc. 12th Int. Conf. Evaluation and Assessment in Software Engineering*, 2008, pp. 71–80.
72. T. Gorschek, P. Garre, S. Larsson and C. Wohlin, Industry evaluation of the requirements abstraction model, *Requirements Engineering J.* **12** (2007) 163–190.
73. I. Sommerville, *Software Engineering*, 8th edn (Addison-Wesley, 2007).
74. V. R. Basili, G. Caldiera and H. D. Rombach, The experience factory, in *Encyclopedia of Software Engineering* (John Wiley & Sons, 1994), pp. 469–476.
75. J. M. Verner and W. M. Evanco, In-house software development: What project management practices lead to success? *IEEE Softw.* **22** (2005) 86.
76. T. Dybå and T. Dingsøyr, Empirical studies of agile software development: A systematic review, *J. Information. Software Technology* **50** (2008) 833–859.
77. M. Weber and J. Weisbrod, Requirements engineering in automotive development: Experiences and challenges, *IEEE Softw.* **20**(1) (2003) 16–24.