

Barriers to Sharing Domain Knowledge in Software Development Practice in SMEs

Jim Buchan¹, Christian Harsana Ekadharmawan¹
¹ School of Computing and Mathematical Sciences,
Auckland University of Technology
Auckland, New Zealand
jim.buchan@aut.ac.nz, christian.harsana@gmail.com

Abstract. The collaborative development of a shared domain understanding between the client stakeholders and the software production team is crucial to the success of software development projects. It is also a challenging and volatile process in practice. There is growing interest in enhancing the development of this shared understanding by improving related processes and support tools. The design and evaluation of such process improvements and tools should be based on robust theory and a clear understanding of the phenomenon and its context in practice. There is however, minimal empirical research on understanding the phenomenon of shared domain understanding in practice in this situation. This paper seeks deeper insights into the process of sharing domain understanding in the context of Small to Medium-sized Enterprises (SMEs) in the software development industry by investigating the barriers and challenges in practice. The study focuses on SMEs because of their economic importance globally and in New Zealand in particular. Small organizations may be especially challenged due to reliance on key individuals and insufficient resource to employ several domain specialists. In this paper we present the results of a field study of commercial software development practice in which we conducted semi-structured interviews with practitioners from ten such organizations. The study provides insights into practices and perceptions related to the challenges software development practitioners face in developing shared domain understanding with the client stakeholders. Our results identify a diverse range of challenges and barriers on which to relate theory and as a foundation for designing process and tool improvement.

Keywords: evidence-based software engineering, knowledge sharing.

1 Introduction and Motivation

The success of a software system is largely based on the degree to which it meets the expectations of a client stakeholder group in terms of addressing some real world problem. The software development process involves the software production team and client stakeholders developing a shared understanding of the problem domain sufficiently well to determine the most appropriate set of features and attributes for a new software system, as part of the solution. This evolution of shared understanding involves a “mesh” of communications and knowledge sharing interactions between the

software production (vendor) team and the client stakeholder group. This includes the client stakeholder group (e.g. users, managers, and domain experts) and the software production group (e.g. analysts, designers and developers) sharing and seeking knowledge using a variety of communication channels and information artefacts. This knowledge is integrated into their existing knowledge or understanding and combined and aggregated to form both teams' understandings of the application domain. Some of this domain understanding needs to be "shared" in some sense to avoid misunderstandings, misaligned expectations and allow cooperative progress. The notion of shared understanding is discussed in more detail in [1, 2] and is taken to refer to the degree of cognitive overlap and mutual beliefs, expectations and perceptions in the following discussions.

Although this domain knowledge sharing is most visible at the elicitation effort early in a software project, this process of articulating, sharing, clarifying, reflecting and sharing understanding is iterative and incremental throughout a software project. As software development progresses, the domain problem is often better understood and the business goals, business requirements and user requirements are consequently refined. The software system goals, requirements and specifications are then also refined to align with this better understanding of the problem domain. Although domain understanding needs to be shared *within* the software development team (and among the client stakeholders), the focus of this paper is the interactions and knowledge sharing across the production team and client stakeholder group boundary.

At the individual level it involves developing an understanding of the application domain, refining that understanding to a level that is appropriate for the role of that individual, and applying it at the time of "need". Thus individuals in the production team and client group are co-dependent to some extent on each other's problem domain knowledge and capabilities to share and internalise this knowledge to suit the individuals' and teams' needs. That evolving individual understanding needs to be periodically shared, "tested", verified and agreed upon so that those involved can work cooperatively towards the same goals that create sufficient value for the clients (and the clients understand this value!).

The sharing of domain understanding is challenging in practice because of the inherent complexities. It may involve a large number of individuals with a broad diversity of existing specialized capabilities, expertise and vocabularies, as well as different cultures, beliefs and values. Their motivations and authority to influence the development project can also be widely divergent. Furthermore the individual and team-level exchanges are characterised by cognitive, social and organisational interactions that are unpredictable and potentially error-prone. This includes, for example, challenging activities such as developing a shared vocabulary, sharing and internalising both conceptually abstract *and* detailed information about the problem domain, reconciling many points of view from diverse stakeholders and accommodating changing and volatile understanding. It also involves periodic verification of some shared representation of the domain understanding and how it maps to the software solution. To add to this complexity, a sufficient level of shared understanding between the two groups can never be certain because of the difficulty in not only defining what is "sufficient" but also in measuring the level of shared understanding. Instead, the misunderstandings, conflict and breakdowns that result from a lack of shared understanding are focused on. This paper takes the approach of identifying the barriers and challenges of sharing understanding, with a view to overcoming them with better tools or techniques.

Although it is inherently challenging, the development of shared understanding is critical to the success of a software project. Successful development of a software system is predicated on the vendor team's understanding of the main concepts, goals and purpose of the software system and how well this aligns with a client group's expectations. A number of researchers in RE (see for example [3-6]) argue convincingly for the central role domain knowledge sharing plays in RE activities. Empirical evidence from their studies of RE practice demonstrates that high quality requirements are crucially dependent on the client and vendor stakeholders sharing a sufficient level of understanding of the problem domain.

The need to understand and improve practice in this area is not lessening either, despite significant advances in modeling, tools and processes over the last few decades. We are seeing the application of software systems to an ever widening diversity of application domain, often conceptually challenging and complex. This broadens and deepens the domain knowledge that developers and other non domain experts have to understand. The need for further research into supporting and comprehending the phenomenon of "developing shared understanding" in practice becomes even more evident in the context of new types of software (e.g. ubiquitous, service oriented, self-managing, or mesh) and emerging development contexts (e.g. global development teams, distributed users, product or market driven development).

There has been considerable research into developing tools, techniques and processes to support the activities and complex interactions that contribute to developing a shared domain understanding in the context of emerging software needs. There seems to be "race" in tool and technique design between knowledge and implementation, however, as pointed out in [7] in the context of Human Computer Interaction design. They argue that "it is a typical pattern in HCI for new ideas to be first codified in exemplary artifacts and only later abstracted into explanations and principles". This pattern also seems to apply to the development of tools and techniques for other aspect of software engineering support. This may be a symptom of the relative immaturity of software development as an engineering discipline. New ideas may be implemented based largely on implicit or tacit knowledge, more of a characteristic of a "craft". As the tacit knowledge is made explicit, and forms the basis of design and implementation of new software development tools and techniques, the discipline is seen as moving more towards "engineering". With this in mind, this study seeks to gain a deeper, explicit knowledge of the challenges of the "phenomenon-in-action" of sharing domain understanding. This should then better support the selection of related theory from disciplines such as cognitive sciences, semiotics, knowledge management and organisational theory. This in turn will help to explain and diagnose the challenges and inform future tool and process design to address them explicitly. It also provides a clear theoretical basis with which to *evaluate* the tools or processes against. Then the purpose of evaluating a new tool or process would be to validate a distributed cognition model or knowledge transfer hypothesis, for example, rather than just understanding the tool. This study is the first part of a larger research project that follows this approach.

In addition, many of the findings and proposed approaches found in existing literature are aimed at large organisations, with the tacit assumption that these findings will apply to small organisation (i.e. having less than 50 employees [8]). This point is highlighted in [9], where the authors argue that RE in small organisations is under-represented in research literature. They further observe that such organisations make up a large part of the software industry) and in [10] they estimate that SMEs contribute 80% to economic growth worldwide. Moreover, it is likely that small organisations are

more vulnerable to the complexities and volatility of developing a shared domain understanding compared to large organisations. It is widely acknowledged in literature that there are some fundamental operational differences between small and large organisations (see for example the Sept/Oct 2000 issue of *IEEE Software*). In empirical studies of small and medium organisations, [11], [12], and [13] characterise them as having fewer resources to devote to tools and hiring domain experts. Compared to their larger counterparts, small organisations appear to be more concerned about practice rather than “compliance” to formal, defined processes. They also observe that small organisations generally focus on shorter term priorities, which are typically directed towards deliverables. These ideas, strengthened by personal observations of small software companies, suggest that current understanding of and approaches to the challenges and barriers to shared domain understanding may not directly transfer to smaller organisations.

This paper addresses this lack of empirical information on challenges and barriers in small organisations in the area of sharing domain understanding, and examines the applicability of previously reported findings, generally drawn from experiences with larger organisations, to small companies. In addition, it is the intention of this research to gain insights into practitioners’ *perceptions* of their practices and challenges in this area. This is based on a desire to “know” the practitioners more deeply as “customers” of research and understand their experiences and needs in this area.

Having provided a brief background to the topic of shared domain understanding in software development, and justified the focus on empirical research in an SME context, the remainder of the paper describes and justifies the design and implementation of the field study (section 2) and describes and discusses the main findings in section 3.

2 Research Design and Implementation

The selection of a research methodology and specific data collection and analysis methods are based on the nature of the research aims and questions. This section describes and justifies these aspects of the research design and discusses the participating organisations and other features of the implementation of the study.

2.1 Aims and Methodology

It is the aim of this research to gain insights into challenges in the development of shared domain understanding in practice through practitioners’ perceptions. The scope of the field study undertaken includes exploring how practitioners conceptualise sharing domain understanding with client stakeholders, the techniques, tools and representations they utilize and their efficacy, and the importance they give to developing this shared understanding. The emphasis of this paper is on understanding the identified barriers and challenges, their causes and consequences, and approaches to addressing these challenges in practice.

In line with other exploratory studies of this type, a multiple case study method, with semi-structured interviews for data collection, is employed. A semi-structured interview was employed, rather than a formal, structured interview or survey, because

it has the advantage of being able to clarify and probe issues and extend the focus of the discussion to interesting aspects *as they arise*. Thus, as observed by [14], a deeper and richer understanding of the phenomenon may be gained. In addition this technique encourages the development of a rapport and trust between the investigator and the interviewee. This is desirable if interviewees are to feel they can freely discuss their practices, challenges and successes.

Note that it is not the intention of this initial study to observe practices or analyse artefacts, which are also common sources of data in case studies. In addition, the viewpoint of the study is restricted to the perceptions of the software vendors, as represented by senior member of the participating software production teams. Comparison of the viewpoints of representatives of the client stakeholder groups is planned for a future study. This paper focuses on presenting the barriers and challenges to shared understanding and discussing the implications for researchers and practitioners.

2.2 Case Organisations

Invitations to participate in this study were sent to 204 organisations, based on the company size (small), and involvement primarily in software development. We selected candidates who had been operating for at least 5 years to allow for maturing of its practices.

Of the candidate organisations invited, 11 organisations initially agreed to participate and 10 organisations actually proceeded with the interviews. Experienced senior-level staff from the organisations were interviewed, with the view that they would have a clear overview of processes as well as some depth of interaction with client representatives, which proved to be the case. Two of the authors of this paper were involved in interviewing all the participants, one facilitating the interview, and the other taking detailed interview notes. The interviews were all recorded on audio tape for later transcription. The interviews were generally located at the place of work, or a neutral venue if requested, and lasted between one and two hours.

The rich and extensive set of data collected from the interviews includes information related to what the important challenges are to developing a sufficient shared understanding with the client stakeholders in practice and why. Thematic analysis of this data was employed as the method of data analysis. As noted in [15] this is a common method of analysing qualitative interview data in order to identify concepts or themes related to a phenomenon. The main construct being analysed is the *process* of developing shared domain understanding. As [16] points out, analysing a process may provide a holistic view of a system of action, which includes activities, roles, artefacts tools and techniques. The unit of analysis is the (small) organisation rather than specific teams or projects.

The participating organisations represent a wide diversity of application domains and include 3 product-driven companies and 7 providing bespoke software development services. All of the organisations had been operating for over 8 years and most of them closer to 20 years. The organisations would be classified as small enterprises with all having fewer than 100 Fulltime Equivalent (FTE) employees. The representatives from the companies were all at a senior level ranging from senior systems analyst to company owners, with 8 of them having more than 10 years'

experience in the software industry and the other two having extensive business or domain experience.

3 Challenges and Barriers to Shared Domain Understanding

This section presents the results of the investigation, discusses implications for practitioners, and speculates on some possible directions for addressing some of the issues.

All organisations had stories of miscommunications and misunderstandings that had contributed to project difficulties and acknowledged a number of challenges and barriers in developing a shared domain understanding. Table 1 depicts the main challenges identified in order of decreasing strength from this study. It should be emphasised that all the barriers are interrelated both causally and hierarchically and are represented in Table 1 according to the strength perceptions of the participants of this study. The relative strength of an identified challenge is estimated on a 10 point scale. It is based on the frequency with which interviewees identified it, the degree of prompting required, and the emphasis they placed on it (voice, body language, anecdotes). The rest of this section discusses the individual barriers presented in Table 2 in more detail, including any root causes and strategies for addressing the challenges identified by the participants in this study. Alignment with results and concepts from existing literature is also discussed.

Table 1 Challenges and Barriers to Shared Domain Understanding

Challenges and Barriers	Relative Strength
Inadequate client representation	10
Inter-group diversity	8
Lack of a common vocabulary	8
Lack of access to key stakeholders	8
Changes in problem understanding	7
Client uncertainty or disagreement	6
Difficult representations of understanding	5
Poor communications practice	3

3.1 Client representation

The challenges presented by poor quality client representatives is consistently emphasised as a significant barrier to shared understanding by all participants. The prevalent view of the interviewees is that the client representative(s) is a key “interface” to the client organisation and if this relationship and interactions are poor then communications suffers and it is difficult to elicit domain knowledge and verify shared understanding. It seemed that this is generally a many-to-one (or few) relationship for the participating organisations. This can be expected to magnify the challenge of a poor representative compared to organisations with multiple points of contact, where a poor quality representative may be outweighed by other high quality representatives.

Characteristics of poor quality client representatives generally related to their perceived lack of domain knowledge, lack of availability or some form of “resistance” to sharing. Table 2 summarises the main challenges identified by the participants, in decreasing order of strength. These are now discussed in more detail including the perceived causes and practical strategies from both the field study and related studies from literature.

Table 2. Challenges Related to Client Representation

Challenge/Barrier Identified
Lack of domain knowledge .
Lack of availability
Actively or passively resistant
Has a hidden agenda.
Poor learner
Indecisive
Lack of external authority
Overly demanding

The most strongly emphasised characteristics of a poor quality client representation relate to perceptions of their lack of domain knowledge. This may be a deficiency in depth or breadth of domain knowledge. It was seen as resulting in an information need not being satisfied (shared) or sharing conceptual understanding being limited. It was also described as resulting in unnecessary uncertainty or volatility in understanding and requirements. Interestingly, “poor domain understanding” was also used to describe client representatives who were poor at articulating their (tacit) knowledge explicitly, or unable to communicate their understanding using terminology and concepts that the software production team could easily “digest”. Also included in this category of “poor domain understanding” were representatives with little understanding of the knowledge or needs of the wider client stakeholder group.

Although it was a strongly identified barrier in this study, the interviewees did not offer any specific explanation regarding why client stakeholders with insufficient domain knowledge may be given the role of client representative. Some clues are found in literature, however. It is suggested in [4] that client representatives may be chosen because of their (high) position rather than domain knowledge. In addition, it is observed in [17] that in small organisations often the most capable (knowledgeable) client stakeholders are too busy contributing to business as usual to be client representatives. Another field study reported in [18] also notes the challenge of client representation with inadequate domain knowledge and concludes that a contributing factor is the lack of involvement of the development team in selecting the client representative(s). Indeed, several participants from our study suggested that this challenge could be alleviated by having more production team control of an identified stakeholder selection process, although it was remarked that clients would be unlikely to “allow” this. Production teams facing this barrier would generally rely on their own in-house domain expertise and would try to share their understanding with the client representative. Participants point out that this is a fairly limited solution, however, since in-house expertise is not always available, client representatives are not always open to “being advised” about their own application domain, and the specific contextual factors related to that particular client organisation, situated in a wider domain ontology, may be missed. It is certainly worth noting that this still seems to be

an important challenge, over a decade after those previously mentioned related studies identified it as so.

Lack of availability of the client representative is also highlighted as being a significant barrier to sharing domain understanding. The discussion on this mirrors that for the challenge of accessing key stakeholders, discussed in detail in section 3.4 and so the reader is referred to that section for more detail.

Another area of inadequate client representation identified as noteworthy relates to problematic attitudes and behaviors of the representatives. Broadly speaking this is perceived as overt uncooperative behavior such as “holding back” feedback or information, or more passive resistance such as always agreeing, with little depth of thought. Active resistance to sharing understanding included refusing to do something requested by the production team. An unwillingness to compromise or negotiate and repeated disagreement on domain understanding were also seen as active resistance. Some participants interpreted a lack of accessibility of the client representative, or overly “secretive” client behavior (inappropriately commercially sensitive), as active resistance also. Passive behaviors such as verifying production team understanding with no challenge or discussion were viewed as less likely to lead to negative conflict, but still regarded as a barrier to shared domain understanding. Similarly, passive attitudes such as a general lack of engagement, or an unwillingness to commit to a position were also viewed as barriers. These concepts identified by the interviewees are close to the notions of “silent resistance” and “compliance resistance” noted in [19] as barriers to shared understanding in requirements engineering.

Participants described such uncooperative behavior as often hindering knowledge elicitation and verification activities throughout the software development lifecycle. They describe the consequences as potentially leading to: delays in the project, misaligned expectations, extra communications effort, lack of buy-in, lack of trust or heightened project risk.

The client representative’s resistance to verifying and sharing understanding was seen as due to a low value being placed on the role by the client representative. This resulted in their lack of “buy-in”, “commitment” or “resentment” to that role. Some participants also noted that certain personality traits, such as selfishness (thinking only of their own gain), could also disrupt communications and sharing understanding. This aligns with the observation in [17] that client stakeholder personalities may be at the root of these attitudinal and behavioral issues. It is suggested in [19] that resistance to the change brought about by the introduction of a new software system may trigger attitudinal or behavior problems in client stakeholders, and this may apply to the client representative.

The interviewees had few suggestions to address this resistant behavior and poor attitudes. One participant suggested that selecting an open-minded, tolerant *production team* representative with good communication skills may assist in modifying the client representative’s behavior. Another participant suggested that support from another client stakeholder with higher authority should be sought.

Power is another clear barrier related to effective communication with the client representative. Some are perceived as playing power games with hidden political agendas, so that aspects of understanding were withheld to the advantage of the client in some way. It also encompasses situations where the client representative introduces their own ideas or agenda, without socializing it first with the other appropriate client stakeholders (particularly their managers). Related to this is the frustration expressed by some participants when trying to negotiate understanding and perhaps reach a

compromise or decide on alternative views, when the representative doesn't have the authority to speak for the organisation and must consult with a higher authority.

It is clear from the interviews that most participants placed significant reliance on getting quality domain expertise and quality feedback and verification of understanding from the client representatives. They generally perceived the selection of the client representative(s) as largely out of their control, although two organisations report influencing the selection of the customer representative through negotiation. Another three organisations reported employing their own domain experts, who act as "proxy" clients. Presumably the client organisations don't actively select a poor quality representative for a project, so the question remains as to how this situation arises? No clear causes are offered by the participants, although the "blame" was certainly placed with the client group by the (vendor) interviewees.

While the quality of client representation is discussed in RE literature ([20], [21]) it appears that for small organisations it is perceived as a particularly significant and frequent barrier to developing shared understanding. Perhaps this is because larger teams in larger organisations have multiple points of contact. This could be a fruitful area for process improvement and better tool support. How can a more visible "client management" process be designed that will promote the selection of the client representatives based on appropriate criteria and support them to engage and commit, despite having competing work pressures? Or how can the knowledge sharing role of the client representative be made less crucial for an SME? One approach may be to support client stakeholders (and members of the production team) to explicate their relevant domain knowledge in a rich representational format that is straightforward to produce, manipulate and verify in a collaborative and distributed mode.

3.2 Inter-group Diversity

"Diversity" between the client and vendor groups is also identified strongly as a barrier to shared understanding. This is described by the participants as being linked to the differences in individuals' characteristics. This includes: their experiences, depth of knowledge, abilities to conceptualise, values, risk tolerance, and priorities. This barrier is conceptualised as resulting in "difference trends" between the groups that developed over a series of inter-group interactions. This can lead to unexpected actions, increased conflict, misunderstandings, miscommunications or misinterpretations that disrupt clear communications and hinder the development of shared understanding. Particularly noted by participants are the differences in depth of knowledge between the two groups, more technical on the software production team side and more business oriented in the case of the client group.

Three particular issues related to inter-group diversity were particularly emphasised by participants. One relates to the "difficulty to get them to speak the same language". This is seen as a significant barrier to developers gaining a sufficient understanding of the business processes and goals, and being the root cause of "some projects failing badly". This is discussed in more detail in the section on the challenges of developing a common vocabulary (with which to create, share and verify understanding).

Another barrier identified related to diversity, is the problem of the software production team "jumping into the coding process before they understand the business goals and processes". This was seen as resulting in a "cycle of change, change,

change”, which is problematic to accommodate and could result in delays. The flip-side of this situation was also reported as a barrier to sharing domain understanding. This is the situation where clients inappropriately, and in the early stages of knowledge sharing, specify aspects of the *solution* system with flawed justification or poor understanding of the solution domain. Participants pointed out that this could restrict problem domain exploration and constrain consideration of alternative candidate domain solutions.

Another interesting challenge perceived by the participants relates to the lack of “big-picture” some client stakeholders exhibit. This is seen as resulting in stakeholders often getting “lost” and “missing the point because they don’t understand what the business is trying to do”. This relates to the notion of “Situation Awareness” (SA) defined in [22], and discussed in terms of inter-team communications in [23]. SA can be described as comprehension of the “big picture” of the elements of a situation. Cognitive psychology suggests that a level of SA is needed for sense-making and decision making regarding the situation, and how it might be in the near future. The high workload of client stakeholders (and possibly production team members) could impair their SA and be an impediment to internalising and sharing understanding. This could result in decision making based on the wrong understanding. This suggests that it may be useful to have some mechanism for developing and sharing the “big picture”, in terms of business goals and aims, for example, and having this front of mind at key decision points in developing shared understanding.

It is worth noting that interviewees did not mention cultural diversity as a barrier to sharing understanding, although it is identified in literature as a common barrier to communications and shared knowledge (e.g. [24, 25]). It may be that the participating organisations may only serve a local market with little cultural diversity.

3.3 Lack of a Common Vocabulary

Although only identified by two organisations during the open questioning phase of the interviews, when prompted all but one participant agreed that a lack of a common vocabulary can be a barrier to sharing domain understanding. They described the issue as the use of “jargon”, or “unfamiliar language” when sharing information for knowledge transfer or validation. Generally this was resolved by repeated clarification and verification interactions. They pointed out that this could be time consuming and cause delays, however. Also, sometimes individuals “didn’t know that they didn’t know” regarding the terminology and by the time they did recognize the confusion the impact on the project was more serious. This issue is also reported in literature and typically the maintenance of a shared glossary is recommended to alleviate the problem [3, 5]. Participants noted that this wasn’t always useful because if the glossary was not always referred to by some client stakeholders (who maybe had no input to it). Other shortcomings given include: the glossary wasn’t clear enough, it had gaps, or it wasn’t up-to-date. Certainly it is restricted in the knowledge represented and provides limited information on concepts and their relationships. Perhaps a shared cognitive map, collaboratively developed and maintained, would serve this purpose better. Of course the overhead in developing and maintaining such a representation of the project “ontology” would need to be minimized in practice.

3.4 Lack of Access to Key Stakeholders

A variety of circumstances are identified as being the root causes of this access challenge including: geographical distance, delegation of responsibility, multiple layers of stakeholders, office policy, high cost of stakeholder involvement, or stakeholder indifference. There is a strong perception that stakeholders (including the client representative) are often too busy with other work (over-worked perhaps). Consequently they can take too long to respond to requests for information or confirmation of understanding, or provide shallow or incorrect responses due to this work pressure. Sometimes this was interpreted as the stakeholder having a lack of commitment or indifference to the project, because they don't "make themselves available".

One of the issues identified by participants with geographically distant client stakeholders is the lack of opportunity for face-to-face interactions and the concomitant rich communications available through this channel. Participants described the use of video conferencing as a partial solution to this. The use of a variety of communication channels (e.g. phone, email, SMS), was a common strategy described. This depended on the nature of the understanding to be shared (e.g. urgency, impact) and the disposition of the particular stakeholders involved, as well as the pragmatics of the situation.

Deficiency in availability often resulted in a lack of scheduled meetings or delays in the schedule, with consequent delays in sharing understanding and project progress. Participants also discussed the need for unscheduled access to certain client stakeholders, particularly the client representative. This was typically for clarification or verification of understanding where the need is urgent and possibly of less significance to the project. This often involved a quick phone call or email. Sometimes, if the required client stakeholder(s) was not available in a timely fashion, members of the production team would act on assumptions, decisions or interpretations that had not been verified with the client stakeholder group, increasing risk and limiting the sharing of understanding.

Multiple layers of stakeholders, for example where a client representative is a third party agent acting on behalf of the client, was also perceived as a challenge. The "thicker" layer of interpretation and possible increased communications times were seen as factors that could increase the chance of miscommunication or delays. One participating organisation report that one particular client had a policy that prevented the software development team from directly communicating with end-users. Although business analysts from the production team could communicate with the end-users, this potentially created a situation similar to the multi-layer stakeholders.

It may be that a persistent representation of some aspects of shared understanding that can be easily manipulated and annotated could provide a partial solution to this barrier by providing an asynchronous, distributed and ideally rich mechanism for sharing understanding. This could lower the accessibility pressure for some key stakeholders.

3.5 Changes in Problem Understanding

A commonly cited issue with requirements management is the difficulty in managing changes to scope and requirements [19]. Although not identified initially by

participants, when prompted about changes to domain understanding, they expressed the view that, such change can be problematic if not “well managed”. They described challenging experiences such as overly frequent changes, clients not sharing changes with the vendor, and lack of clarity on the wider impact of new understanding. Overall they saw the changes in understanding as positive and a natural part of the evolution of shared understanding.

3.6 Client Uncertainty or Disagreement

Over half of the participants indicated that client uncertainty with the problem domain is a barrier to sharing domain understanding. This is subtly different to insufficient domain knowledge. The client stakeholders may have sufficient depth of understanding but they are uncertain about which aspects of their domain knowledge apply or are important. A factor in this uncertainty may be that different stakeholders “compete” for their views to be the prevalent shared understanding.

Uncertainty with the envisioned the system goals was also seen as a possible barrier. While this is expected at the project concept stage and early phase requirements, some participants observed that this uncertainty could be a repeated pattern of and hamper decisions about shared understanding.

Another challenge to sharing domain understanding was described as the situation where there is a low overlap of some areas of understanding within the client stakeholder group. This may manifest as inconsistent, conflicting or competing points-of-view being “shared” by the client group. Participants observed that the client stakeholders may not be aware of this situation because they have had no need (or opportunity) to integrate or share these aspects of their understanding with each other before. This barrier is well acknowledged in literature and has been identified in other empirical studies such as [20, 26].

Most participants suggested that this challenge could be addressed by finding a mechanism to facilitate the client stakeholders to agree on important aspects of the problem domain, and “achieve buy-in” to these shared views, *before* they share this understanding with the software production team. Participants acknowledged that this is generally a challenge in practice. In [20] it is suggested that when the software team identifies the divergent client views they should organize a meeting with all the related (conflicting) client stakeholders together and facilitate a shared view. While this is not uncommon during early requirements elicitation, it is often difficult to get such a group together again in one place at one time because of divergent work schedules. A distributed, virtual mechanism of knowledge sharing that improves the visibility of all relevant clients stakeholders’ points of view may assist with overcoming this barrier. It would have to be low effort and low complexity to be practicable, however.

3.7 Difficult Representations of Understanding

Participants observed that often the sharing of domain understanding involved the development and sharing of representations of aspects of the knowledge to be shared and verified. The representations identified by interviewees as representing shared

domain knowledge includes formal and structured representations as well as: flow charts, business process diagrams, scenarios, use-cases, requirements specification documents, various UML diagrams, conversations and email threads. Screen shots, prototypes and product demonstrations were also identified as domain knowledge representations, with the reasoning that contain “embedded” (but constrained) understanding of the problem domain, and that they uncovered misunderstandings.

Participants perceive the use unfamiliar or overly complex representations as a barrier to sharing understanding. It is effectively introducing a new vocabulary that is not common to the two groups, as previously described. Interviewees emphasised the importance of being aware of the clients’ fluency in interpreting and manipulating different knowledge representations and notations. This is in line with the findings of [14, 17, 26] who link the use of certain knowledge representations (UML, object oriented representations and use cases) with barriers to knowledge sharing. These studies also suggest that pictorial and concrete representations of knowledge (e.g. screen shots or prototypes) are typically easier for the client stakeholders to understand. This is supported by the experience of the participating organisations.

Natural language (both semi-structured and unstructured) was identified as the most common representation of shared understanding because it is the “lowest common denominator” for understandability. It was also noted that the ambiguity inherent in natural language is a source of challenge to sharing understanding and difficult to manage, as also reported in [14, 27] in relation to requirements specifications. The mechanism for discovery of ambiguity in natural language knowledge representations typically involved reviews, cross validation against other representations and frequent clarification interactions. One participant had a strong conviction that written communications “in any form” are always ambiguous and would never be a substitute for regular face-to-face meetings where more signs are available to identify ambiguity and instant clarification is possible. [11] suggests that being able to detect ambiguity and imprecision in natural language is a skill that that can and should be learned.

A few participants noted that poor presentation of knowledge representations can be a barrier to shared understanding. They described “unattractive”, “dry”, “dull”, and “voluminous” documents for review or confirmation of understanding as demotivating and limiting client engagement with the knowledge sharing activity.

3.8 Poor Communications Practice

Communications and sharing understanding are closely related [4] and so poor inter-group communications are strongly linked to challenges in developing shared understanding. The communications barriers identified by interviewees generally related to deficiencies in the timeliness or frequency of the communications, or a lack of “rich” communications interactions. These challenges were seen as arising from insufficient communications planning, lack of stakeholder availability (for all the reasons previously discussed), unrealistic project timeframes, or the use of project processes or methodologies that de-emphasised client communications. Poorly defined or socialised roles and responsibilities are also perceived as a source of challenge to communications practice. This is one of the factors identified in [4] also.

Clearly sharing understanding benefits from the availability of a diversity of communications channels, knowledge artefacts and communication related roles. Multi-view models of communications interactions, such as that introduced by [28] may provide fuller understanding of the effectiveness of communications for sharing understanding.

4 Conclusion

Overall, participants identified a broad range of interrelated barriers and challenges to adequate sharing of domain understanding. The potential for client representatives to inhibit the sharing of domain understanding between the two groups was emphasised in the frequency and strength with which interviewees, unprompted, raised this as an issue. The next most forcefully expressed barrier related to the diversity in “world views” and experiences of the vendor and client groups. This contributed to a number of communications issues that obstructed sharing understanding between the two groups.

This study achieved its aim of deepening the understanding of the barriers and challenges faced by software development teams in collaboratively sharing domain knowledge with the client stakeholder group. This will now be used to guide the development of multi-view models of sharing domain understanding in this context. Theories from cognitive science, organisational theory, semiotics and knowledge management will be considered for candidate principles to inform the models. The longer term plan is to then test the models through the design and evaluation of tools, techniques and processes based on these models.

References

- [1] J. A. Cannon-Bowers and E. Salas, "Reflection on shared cognition.," *Journal of Organizational Behavior*, vol. 22, pp. 195-202, 2001.
- [2] S. G. Cohen and C. B. Gibson, "In the beginning: Introduction and framework," in *Virtual teams that work: Creating conditions for virtual team effectiveness*, C. B. Gibson and S. G. Cohen, Eds. San Fransisco: John Wiley & Sons, 2003.
- [3] E. G. Alcázar and A. Monzón, "A process framework for requirements analysis specification," in *Proceedings of 4th International Conference on Requirements Engineering*, 2000, pp. 27-35.
- [4] J. Coughlan, M. Lycett, and R. D. Macredie, "Communication issues in requirements elicitation: a content analysis of stakeholder experiences," *Information and Software Technology*, vol. 45, pp. 525-537, 2003.
- [5] R. Offen, "Domain Understanding is the Key to Successful System Development," *Requirements Engineering*, vol. 7, pp. 172-175, 2002.
- [6] A. Osada, D. Ozawa, H. Kaiya, and K. Kaijiri, "The role of domain knowledge representation in requirements elicitation," in *25th IASTED International Multi-Conference Software Engineering*, Innsbruck, Austria, 2007, pp. 84-92.
- [7] A. G. Sutcliffe and J. M. Carroll, "Designing claims for reuse in interactive systems design," *International Journal of Human-Computer Studies*, vol. 50, pp. 213-241, 1999.

- [8] EU, "The new SME definition. User guide and model declaration. ," European Commission, 2005,
http://www.ec.europa.eu/enterprise/enterprise_policy/sme_definition/sme_user_guide.pdf
- [9] J. Aranda, S. Easterbrook, and G. Wilson, "Requirements in the wild: How small companies do it," in *Proc. 15th IEEE International Requirements Engineering Conference RE '07*, 2007, pp. 39--48.
- [10] S. Pavic, S. C. L. Koh, M. Simpson, and J. Padmore, "Could e-business create a competitive advantage in UK SMEs?," *Benchmarking: An International Journal*, vol. 14, pp. 320-351, 2007.
- [11] E. Kamsties, K. Hörmann, and M. Schlich, "Requirements engineering in small and medium enterprises," *Requirements Engineering*, vol. 3, pp. 84-90, 1998.
- [12] F. Pino, F. García, and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review," *Software Quality Journal*, vol. 16, pp. 237-261, 2008.
- [13] A. Atherton, "The uncertainty of knowing: An analysis of the nature of knowledge in a small business context," *Human Relations*, vol. 56, pp. 1379-1398, 2003.
- [14] L. Karlsson, A. G. Dahlstedt, B. Regnell, J. N. Dag, and A. Persson, " Requirements engineering challenges in market-driven software development - An interview study with practitioners," *Information and Software Technology*, vol. 49, pp. 588-604, 2007.
- [15] M. Miles and A. Huberman, *Qualitative Data Analysis*. Sage, CA: Thousand Oaks 1994.
- [16] W. Tellis, "Application of a case study methodology," *The Qualitative Report*, vol. 3, 1997.
- [17] K. E. Emam and N. H. Madhavji, "A field study of requirements engineering practices in information systems development," in *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, 1995, pp. 68-80.
- [18] A. Al-Rawas and S. Easterbrook, "Communication problems in requirements engineering: A field study," in *1st Westminster Conference on Professional Awareness in Software Engineering*, London, 1996.
- [19] H. Saiedian and R. Dale, "Requirements engineering: Making the connection between the software developer and customer," *Information and Software Technology*, vol. 42, pp. 419-428, 2000.
- [20] L. Cao, & Ramesh, B., "Agile requirements engineering practices: An empirical study," *IEEE Software*, vol. 25, pp. 60-67, 2008.
- [21] C. Lu, Chu, W. C., C. Chang, and C. Wang, "A model-based object-oriented approach to requirements engineering (MORE)," in *31st Annual International Computer Software and Applications Conference*, 2007, pp. 153-156.
- [22] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors* vol. 37, pp. 32-64, 1995.
- [23] C. A. Bolstad, P. Foltz, M. Franzke, H. M. Cuevas, M. Rosenstein, and A. M. Costello, "Predicting situation awareness from team communications," in *51st Annual Meeting of the Human Factors and Ergonomics Society*, Santa Monica, CA, 2007.
- [24] D. Damian, "Stakeholders in global requirements engineering: Lessons learned from practice," *IEEE Software*, vol. 24, pp. 21-27, 2007.
- [25] Y. Hsieh, "Culture and shared understanding in distributed requirements engineering," in *International Conference on Global Software Engineering*, 2006, pp. 101-108.
- [26] M. Lubars, Potts, C., & Richter, C. (1993). A review of the state of the practice in requirements modelling. *Proceedings of IEEE International Symposium on Requirements Engineering*, 1993, 2-14.
- [27] M. Gervasi and D. Zowghi, "Reasoning about inconsistencies in natural language requirements," *ACM Transactions on Software Engineering and Methodology*, vol. 14, pp. 277-330, 2005.
- [28] J. Aranda and S. M. Easterbrook, "Distributed Cognition in Software Engineering Research: Can it be made to work," in *First workshop on Supporting the Social Side of Large Scale Software Development (SSSLSSD)* Banff, Alberta, 2006.