



Reformulating Requirements Modelling for Digitalisation: A Structuration and Semiotic Informed Approach

Adrian Benfell^(✉) and Zoe Hoy

University of Portsmouth, Richmond Building, Portland Street,
Portsmouth PO1 3DE, UK
{adrian.benfell, zoe.hoy}@port.ac.uk

Abstract. Articulated within this paper is a qualitative investigation into using structuration and semiotic theories that suggests improvements to communication when developing new software. Forming the analysis, an Information System (IS) designed using the Unified Modelling Language (UML) allied with the Agile software development process exposed a key factor: the need for effective communication. This factor caused a negative outcome when developing a new IS necessary to support the digitalisation of a business. Analysing the conditions and adjusting requirements modelling with structuration and semiotic theories, team members expressed with clarity requirements for their new IS as real-world effects and causes. Such cause and effect statements align to Peircean interpretant signs that facilitated effective communication.

Keywords: Structuration · Semiotics · Requirements modelling
Digitalisation

1 Introduction

Digitalisation, the digital transformation of business, requires businesses to be more agile, people-orientated, pioneering, and customer-centric. Some articles point to digital technologies and their transforming properties, for example [4, 19, 22, 41]. Extracting key themes from [4, 19, 22, 41] suggests that effective communication is a requisite in any digitalisation process that needs new software. To realise digitalisation however, some persistent challenges must be answered. For instance, [15, 34, 36] draw attention to the limitations of requirements modelling for software development. They highlight that communication inconsistencies between people when embarking upon requirements modelling can cause negative project outcomes. Hence, in this paper we focus upon improving communication in relation to software development.

Based upon our experience with an Information Systems Development (ISD) project, we argue that when using the Unified Modelling Language (UML) [42] combined with Agile [1], as contemporary requirements modelling tools, do not promote effective communication during software development. Our view follows [5, 10] who emphasise that communication underpins requirements modelling. Many researchers place an emphasis upon effective communication when requirements modelling, however few

provide empirical evidence to resolve this challenge. In this paper we take a similar position to [5, 10] and judge communication to be a permanent and relevant issue when requirements modelling.

Using UML as an up-to-date example, challenges to effective communication during requirements modelling start with UML Use cases. Modelling them further with UML Activity diagrams and narrative to describe functional properties changes the dynamics of communication between project stakeholders. Hence, communication focuses upon Use cases to depict state changes between the functional elements of IS and end-users. Extracting functionality from Use cases with Activity diagrams, the three stereotype classes, entity, boundary and controller, as they comply with the model-view-controller architectural design pattern [23], stabilise requirements when UML Communication modelling early in the requirements modelling process. Also, at the start of UML Communication modelling to show how stereotype classes communicate with each other, complexity increases when detailing requirements with full Object-Orientated (OO) class notation.

We assert, that during requirements modelling, intricacies that materialise when using UML modelling techniques to specify IS functionality impedes effective communication. In this paper we use [3, 12, 26, 30, 33] who define modelling techniques as abstract representations of the problem domain using narrative combined with diagrams. Modelling techniques format communication during ISD, hence recognising how communication works in organisations should become a primary mechanism to facilitate improvements in requirements modelling. This view takes us to structuration and semiotic theories and frames the research questions (RQ) in this paper as:

RQ1: What can structuration and semiotics inform about communication when requirements modelling?

RQ2: How can structuration and semiotics reformulate requirements modelling to improve communication?

To answer the RQs, the structure of this paper is as follows. The next section, theoretical background, presents structuration and semiotic theories as they relate to requirements modelling and acts as preparation to answer the RQs. Following the theoretical background, the research method section includes a case study as supporting empirical evidence derived from an ISD project. The theoretical background provides the scope to assess the case study. Also, to fit digitalisation, the research method section includes recommendations to resolve communication issues when requirements modelling. This paper finishes with a conclusion that discusses outcomes and further research.

2 Theoretical Background

Giddens [16–18] defines the relationship between structure (the rules and resources that reproduce social systems) and agency (the conduct of an individual and of others) as structuration theory. A broad range of viewpoints exists regarding structure and agency. First, Durkheim [13] who emphasised stability and durability of structure, and second, Marx [28] who defined structures as protecting the few and doing little to meet

the needs of many. Giddens [17] however places structure at different levels within society. At the top-level, society comprises socioeconomic stratifications, for example social class. On a mid-level scale, institutions and social networks, and at a lower-level, a community or a grouping of social norms that outlines agency. By developing structuration theory, Giddens defines the ‘duality of structure’, whereby social structure is both the medium and the outcome of social action and people (as agents) share an ‘equal ontological status’.

Giddens’s framework offers three parts to structuration theory. First, signification, where signs used in communication convey meaning specific to organisations. Communication reference points [11], for example police uniforms, acronyms in professional languages and so on, are organisation specific signs. Charlton and Andras [11] also suggest that specialised communication reference points demarcate organisations from others. Second, legitimation includes normative perceptions rooted as social norms that govern people in their actions and understanding in organisations. Third, domination, links to how individuals in influential roles apply power to control resources also in organisations. Hence, the mix of signification, legitimation, and domination links to semiotic theory which are discussed next.

2.1 Signification

Giddens [17], whilst taking a semiotic view of signs and their interpretation, provides little guidance about semiotics. However, semiotics includes two strands, one by de Saussure [35] and the other by Peirce [32]. Both strands have a subjective element to understanding signs present in communication, as signs can have different meanings in different organisations. However, Saussurean semiotics focuses upon communicative acts and text only [14] but Peirce’s view of semiotics helps to comprehend signs used in communication and requirements models. For example, to understand a sign Peirce and Saussure defined the term semiosis, the process of interpreting signs based upon different elements. In Peircean semiotics, the interaction between three combined parts: a sign, an object and an interpretant forms semiosis. A sign may align to three types: symbolic, iconic, and indexical. Regarding Fig. 1, symbolic signs identify their objects through cultural conventions such as words; iconic signs represent their objects by similarities, such as images; and indexical signs have causal links to their objects, for example smoke identifying fire.

Regarding the process of semiosis in Fig. 1, a sign refers to an object, and the object verifies its sign. Therefore, the object is the semantic meaning of a sign due to the ontological association. Interpretant signs add to the sign – object association, as an interpretant is a more developed sign in someone’s mind. Hence, the interpretant gives the contextual meaning given to a sign and may comprise social norms. Giddens view of legitimation and domination identifies how social norms provide the rules by which meanings emerge as interpretant signs. Also, according to Liu [24], semiosis is a sign-mediated process in which people give meanings to signs.

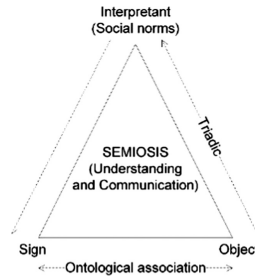


Fig. 1. Semiotic triangle, the sign – object association and interpretant signs

2.2 Legitimation

The interaction of agent and structure, as a system of normative behaviour supports structuration. Legitimation is applicable when attempting to understand communication issues when requirements modelling. For example, Giddens [17, 18] uses the term reflexivity to describe normative behaviour as cause and effect cycles that occur when people revise their social world and alter their place in structures. In this setting, the interaction of people with structures induces normative behaviour. The interaction depicts how people understand their environment through an array of influences, marking out behavioural patterns that when followed define social normative behaviour [40]. Hence, social norms evolve through the interaction of people when following expected conventions within organisations [7]. Social norms emerge as legalised constraints in a contract, or surface resulting from reflexive social interaction [17].

2.3 Domination

Charlton and Andras [11] and Luhmann [27] suggest that previous dialogue forms communication reference points that sustain contact whilst shaping ongoing discourse. People when they belong to an organisation have conversations using specific communication reference points. Thus, organisation specific communication reference points help to distinguish social norms owned by one organisation from another. An influence on social norms includes domination. Identified within structuration theory, domination identifies how people exert control over the rules and resources that influence social norms. However, through reflexivity, people may accept and change social norms shaped in power relationships. Domination thus influences understanding of social normative behaviour as interpretant signs.

3 Case Study

To answer the RQs, we present empirical evidence in a case study format split into four sections: (3.1) overviews the context of the study; (3.2) outlines the ISD challenge based upon UML and Agile; (3.3) details an observation and analysis from the viewpoint of the theoretical background and answers RQ1; (3.4) includes proposals to reformulate requirements modelling to improve communication answering RQ2.

3.1 Project Context

Based upon a digitalisation project that started in 2013, its remit was to develop and deploy an IS to manage accommodation. The team comprised seven support managers, a senior support manager and a head of residential services. The team was self-managed and where possible involved team members in many stages of the Agile approach. Related to Information Technology (IT), all team members showed competence when using known computer-based office products. Team members provided feedback comments only and did not take part in software coding.

Provision for the project came from a service department offering computing support, of which two further staff members helped to scope initial project requirements. One author of this paper involved with the ISD project took the title requirements engineer, whilst the other provided expertise associated with Agile. Working with the accommodation management team, the requirements engineer carried out requirements modelling and software development, organised and attended review meetings, and catalogued feedback. Access granted to historical hard copies of documentation associated with core business processes identified the ISD project context.

The project included one year for ISD, one year for a post implementation review, and a third year (and a fourth if required) for any modifications. The project took three-and-a-half years to complete. Additional supporting data collection methods included documentation associated with the ISD project as versioned requirements models, meeting agendas, and recorded minutes. Interviews were unnecessary as ample information, assembled by the requirements engineer, describes the situation.

3.2 An Outline of the ISD Challenge

The approach taken to develop the new IS, at the request of the computing department, asked that the project to undergo a full systems analysis and design strategy to suit the needs imposed by ITIL and its Service Design guidelines [21]. UML selected as a standardised set of modelling techniques suited ITIL, and UML Use case modelling shaped requirements modelling prior to systems design in an iterative ISD process cycle following Agile principles. In addition, to drive requirements modelling, UML Activity and Communication modelling techniques aligned to Use cases and Agile User stories, baselined requirements. The Agile approach followed the most popular Scrum VersionOne [43] that helped create an iterative and incremental approach to ISD. Meetings, iterations, and iteration planning according to VersionOne [43] formed the generalised ISD process. Iteration planning had a longer-term outlook and determined the high-level requirements as features needed for the new IS. User stories identified the Use cases which helped capture initial requirements, following the concept introduced by [6].

For the plan-driven approach taken, a Computer Aided Software Engineering (CASE) tool facilitated requirements modelling. User stories included a small amount of narrative, one paragraph only, to help create a shared understanding between all team members [31]. At the end of each iteration, team members provided feedback informing any changes needed [20]. With all team members in different locations presented challenges. This included working different shift patterns, and distances

travelled. As a result, this reduced the number of opportunities to have consistent communication with all team members.

The versioned requirements models produced from UML modelling techniques informed initial and detailed class design as development iterations. For implementation, a completed class model coupled with relational database design advised software coding specifications. Also, it was compulsory that the class and relational database models complied with the model-view-controller design pattern to suit a client-server architectural setting for multi-user access. The database installed on the server (as the model), controller classes programmed to handle data transactions in Structured Query Language (SQL) commands across a network, and the views constructed to conform to the underlying data structure, followed the needs of all team members.

The planned Agile software development process linked to the UML Use case driven approach had a supporting project lifecycle and agreed milestones. Sanctioned by all team members, a project initiation document detailed the overall scope of the ISD project and included a short and precise narrative explaining the outcomes. Accomplished in team meetings and by visiting each team member, Agile User stories and linked UML Use cases formed the initial baselined requirements. UML Use cases underwent further development using UML Activity and/or Communication diagrams. UML Communication models aligned to Use cases helped to generate a final Communication model in the CASE tool. Applying UML stereotype classes in the final UML Communication model assisted in designing classes to fulfil the model-view-controller architecture. This approach intended to formalise requirements before moving to software coding.

3.3 Observation and Analysis

After achieving an initial agreement for Agile User stories and associated UML Use case model, complications surfaced. Information encapsulating reflexivity as social norms against communication reference points in an interpretive scheme (signification), proved difficult to capture using UML modelling techniques. To ensure that the new IS met with success, we deemed this information essential to realise the needs of each team member and match it over to the information required to create class diagrams to remove complications. The attempt to capture and represent this information resulted in numerous modifications to the UML Use case model, Agile User stories, and UML Activity and Communication diagrams but without success. Hence, the project ran into severe time difficulties.

To capture requirements for the new IS implied that previous dialogue formed communication reference points, which we did not expect, for the basis of contact between team members whilst shaping on-going discourse based upon social norms, aligning to [11]. In addition, the requirements engineer intervened attempting to keep the project on target as many of the Agile User stories became invalidated as the project advanced through requirements modelling towards software coding. However, team members identified and agreed upon real-world effects. For example, with the existing IS one team member stated, “we produced more accurate case reports when updating case details with correct time allocations”. Team members detailed effects that occurred in the real-world and agreed to them, and in doing so, identified what we described as social norms linked to communication reference points.

When discussing functionality with Agile User stories and UML Use cases, team members lost the link between the functional description and what they wished to achieve. However, a prototype relational database designed from an existing, but incomplete, requirements model helped meet the initial one-year deadline. This plan eased time pressure, but the prototype required further development. A review into how requirements modelling could meet the fast-changing demands placed by team members allowed us to consider how communication reference points, linking to social norms, could reformulate requirements. The idea was to generate requirements for the project as short and precise statements. First signifying communication reference points in a process of semiosis. Second, aligning reflexive cause and effect cycles to Peirce's view of sign and object. Third, using legitimisation and domination to identify social norms as Peirce's interpretant signs.

3.4 Reformulating Requirements Modelling

To represent communication reference points, we identified a suitable scheme that provided a means to categorise information harvested during requirements modelling. Peircean semiotics, further explained by [2, 9, 37, 38] provided a way to adjust requirements modelling using symbolic, iconic and indexical signs. First, symbolic signs as particular words structured what we called Normative Statements based upon Fig. 2. Symbolic signs included the rules that governed how words formed phrases and sentences within each information element (such as Project, Project Scope and so forth in Fig. 2). Second, icons as oval shapes with names identifying UML Use Cases and linked Normative Statements. Third, indexical signs depicted the association between communication reference points and social norms, and between causes and effects structured as narrative declarations. Hence, Normative Statements modelled reflexivity as cause and effect cycles understood and communicated by team members. To ensure consistency when deciding cause and effect statements, we followed necessary and sufficient conditions [8]. Also, each cause statement that supported an effect became a candidate function within a candidate class. Normative Statements as we designed them, codified interpretant signs.

Normative Statements for each UML Use case emerged by collective agreement across connected team members as Roles (shown in Fig. 2). For revision to requirements modelling, each Normative Statement promoted effective communication. For example, Normative Statements allowed an autonomy of expression with short natural language blocks of text [34] that allowed team members to detail supporting necessary conditions to real-world effects. They also captured a balance between agency and structure through the principle of legitimation and domination. Based upon Fig. 2, Normative Statements forming a complete requirements model emerged within two months, four months shorter than when using UML, and reflected the demands placed by team members.

Using Normative Statements helped us to meet the original ISD project timeframe. This approach also removed the need for further modelling with UML Activity and Communication diagrams. We claim Normative Statements simplified requirements modelling, improved communication, and validated the use of structuration and semiotic theories to understand communication issues.

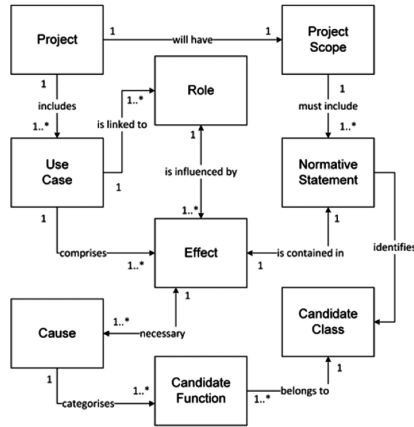


Fig. 2. The structure of normative statements and codification of interpretant signs

4 Discussion and Conclusion

Structuration and semiotic theories provided a method to understand and modify communication when requirements modelling. For example, signification identified the use of communication reference points in communication that link to social norms based upon legitimation and domination. Combined, these items demarcate organisations and semiotic theory helped to represent this phenomenon. Thus, communication reference points in the form of Normative Statements acted as a prime mechanism by which team members understood and communicated requirements for the ISD project.

Underpinned by necessary and sufficient conditions [8], supporting real-world effects structured by Fig. 2 are our view of Peircean interpretant signs. This approach considered how words and phrases, structured inside Normative Statements, combined to represent meaning attached to them by team members for requirements [34]. The necessary and sufficient conditions allowed team members to understand and communicate reference points and social norms.

Using structuration and semiotic theories as the focal point for research, generalisations are possible with similar situations in organisations. For further research, testing the propositions in this paper needs to advance by considering design artefacts and attaching their composition to Fig. 2. Although UML and Agile are useful to develop software artefacts that support IS, our recommendations suggest that an alternative way of requirements modelling incorporating Giddens structuration theory and Peircean semiotics is plausible to meet communication challenges, and therefore digitalisation. However, semiotics needs more research in a digitalisation context.

Stamper [39]; Liu [24] and Liu and Li [25] place an emphasis on studying the effect of signs in organisation contexts. Their view of semiotics forms an inquiry system approach to analysing organisations from the viewpoint of sign creation, utilisation and meaning that govern the behavioural aspects of interaction. Stamper devised a ‘Semiotic Ladder’ as an Organisational Semiotic (OS) approach to investigate organisations based upon Morris [29], shaping most of the work with the realm of OS.

Stamper's Semiotic Ladder distinguishes between organisational and IT domains. It says the IT domain supports organisations and gives technology a particular flavour according to 'social worlds'. Social worlds comprise beliefs, law and culture feeding into Giddens legitimation and domination as intentional communication. OS offers a wealth of research to develop further Normative Statements.

References

1. Agile: Agile Alliance (2018). <http://www.agilealliance.org/>. Accessed Dec 2018
2. Atkin, A.: Peirce's Theory of Signs (2013). <http://plato.stanford.edu/archives/sum2013/entries/peirce-semiotics/>. Accessed Nov 2016
3. Avison, D., Fitzgerald, G.: Information Systems Development: Methodologies, Techniques and Tools. McGraw Hill, New York (2006)
4. Bharadwaj, A., El Sawy, O., Pavlou, P., Venkatraman, N.: Digital Business Strategy: Toward a Next Generation of Insights (2013)
5. Bano, M., Zowghi, D.: A systematic review on the relationship between user involvement. *Inf. Softw. Technol.* **58**, 148–169 (2015)
6. Beck, K.: Extreme Programming Explained. Addison-Wesley, Boston (2000)
7. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. *Comput. Math. Organ. Theory* **12**(2–3), 71–79 (2006)
8. Brennan, A.: The Stanford Encyclopaedia of Philosophy (Winter 2012 Edition) (2012). <http://plato.stanford.edu/archives/win2012/entries/necessary-sufficient/>. Accessed June 2015
9. Chandler, D.: Semiotics – The Basics, 1st edn. Routledge, Abingdon (2002)
10. Charaf, C.M., Rosenkranz, C., Holtén, R.: The emergence of shared understanding: applying functional pragmatics to study the requirements development process. *Inf. Syst. J.* **23**, 115–135 (2013)
11. Charlton, B., Andras, P.: The modernization imperative. Imprint Academic, Exeter (2003)
12. Chen, P.: The entity-relationship model - toward a unified view of data. *ACM Trans. Database Syst.* **1**, 9–36 (1976)
13. Durkheim, E.: Socialism. Collier-Macmillan, New York (1962)
14. Eco, U.: A Theory of Semiotics. Indiana University Press, Bloomington (1976)
15. Fernández, D.M., Wagner, S.: Naming the pain in requirements engineering: a design for a global family of surveys and first results from Germany. *Inf. Softw. Technol.* **57**, 616–643 (2015)
16. Giddens, A.: Central Problems in Social Theory: Action, Structure, and Contradiction in Social Analysis. University of California Press, Berkeley and Los Angeles (1979)
17. Giddens, A.: The Constitution of Society: Outline of the Theory of Structuration. University of California Press, Berkeley and Los Angeles (1984)
18. Giddens, A.: Modernity and Self-Identity, Self and Society in the Late Modern Age. Stanford University Press, Palo Alto (1991)
19. HBR Analytics Services: Accelerating the Pace of Digital Transformation; Harvard Business Review (2016)
20. Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S.: A systematic literature review on agile requirements engineering practices and challenges. *Comput. Hum. Behav.* **51**(2015), 915–929 (2015). <https://doi.org/10.1016/j.chb.2014.10.046>
21. ITIL: Best Practice Solutions (2015). <https://www.axelos.com/best-practice-solutions/itil/what-is-itil>. Accessed Nov 2015

22. Kane, G.C., Palmer, D., Phillips, A.N., Kiron, D., Buckley, N.: Strategy, Not Technology, Drives Digital Transformation. MIT Sloan Management Review and Deloitte University Press, July 2015
23. Krasner, G.E., Pope, S.T.: A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *J. Object Oriented Program.* **1**(3), 26–49 (1988)
24. Liu, K.: *Semiotics in Information Systems Engineering*, 1st edn. Cambridge University Press, Cambridge (2000)
25. Liu, K., Li, W.: *Organisational Semiotics for Business Informatics*. Routledge, Abingdon (2014)
26. Loucopoulos, P., Karakostas, V.: *System Requirements Engineering*. McGraw-Hill, Inc., New York (1995)
27. Luhmann, N.: *Introduction to Systems Theory*. Polity Press, Cambridge (2012)
28. Marx, K.: ‘Preface’, to a contribution to the critique of political economy. In: Marx, K., Engels, F. (eds.) *Selected Writings*. Lawrence and Wishart, London (1968)
29. Morris, C.W.: Foundations of the theory of signs. In: *International Encyclopedia of Unified Science*, pp. 1–59. Chicago University Press (1938)
30. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*, pp. 35–46. ACM, May 2000
31. Patton, J.: *User Story Mapping: Discover the Whole Story, Build the Right Product*. O’Reilly, Sebastopol (2014)
32. Peirce, C.: *The Essential Writings: Selected Philosophical Writings (1893-1913)*, 1st edn. Indiana University Press, Bloomington (1998)
33. Pohl, K.: *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer, Heidelberg (2010)
34. Rosenkranz, C., Corvera Charaf, M., Holten, R.: Language quality in requirements development: tracing communication in the process of information systems development. *J. Inf. Technol.* **28**, 198–223 (2013)
35. de Saussure, F.: *Course in General Linguistics* (trans. R. Harris). Duckworth, London (1983)
36. Schwarz, A., Chin, W.W., Hirschheim, R., Schwarz, C.: Toward a process-based view of information technology acceptance. *J. Inf. Technol.* **29**, 73–96 (2014)
37. Short, T.: *Peirce’s Theory of Signs*, 1st edn. Cambridge University Press, Cambridge (2007)
38. Sowa, J.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole Thomson Learning, Pacific Grove (2000)
39. Stamper, R.: Signs, information, norms and systems. In: *Signs of Work*, pp. 349–399 (1996)
40. Thøgersen, J.: Social norms and cooperation in real-life social dilemmas. *J. Econ. Psychol.* **29**, 458–472 (2008)
41. UK GOV (2017). <https://www.gov.uk/government/publications/made-smarter-review>. Accessed Feb 2018
42. UML, 2.5: Unified Modelling Language Resource Page (2018). <http://www.omg.org/spec/UML/2.5/PDF/>. Accessed Dec 2018
43. VersionOne: 10th Annual state of agile survey (2016). <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>. Accessed Dec 2016