The London School of Economics and Political Science

To Be or To Become? An Enquiry into the Changing Nature of Requirements in Open Source Health IT

Daniel Adrian Curto-Millet

A thesis submitted to the Department of Management of the London School of Economics and Political Science for the degree of Doctor of Philosophy, London, April 2014

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

I declare that my thesis consists of 92000 words.

Abstract

This thesis develops a contemporary problematisation of software requirements. It departs from traditional conceptions of requirements as simple, tamed objects with deterministic force over socio-technical actors and based on assumptions of stability. Such views can lead to a narrow, ultimately unfruitful understanding of the significance of requirements and denied wider consequences of their modes of articulation. Instead, the thesis builds on perspectives where requirements are complex and interactive actors.

The thesis uses openEHR—an open source health IT project aiming to build interoperable Electronic Health Records (EHRs)—as a case study. Studying open source practice offers a good opportunity to consider the nature of requirements because there is an ongoing debate about requirements' role and influence on development activities and project organisation.

The analysis uses Deleuzian concepts of assemblage, multiplicity and becoming. These themes align with a larger body of work influenced by STS and process oriented theorisations, which see the world as dynamic and performative. The philosophy of Deleuze and Guattari in particular provides a counter-balance to any assumed stability in the world.

The thesis presents a new account of the nature of requirements, one that reflects their complex entanglement within software development and open source in particular. Requirements are not insipid descriptive statements that abstract and simplify the world deterministically. They have an intricate existence which serves to hold the potential in the assemblage to become many things. In particular, requirements insinuate themselves into a project's identity, guide a project through

territories—some to be explored, some to be disregarded—and demand specific ways to be recognised, engaged, and cared for. The thesis argues that requirements are more virtual than originally thought, having a subtle, not necessarily visible influence on their assemblages and the way socio-technical actors can potentially relate to the project itself.

Acknowledgements

The first person I would like to thank is Almudena, my wife. She is the reason this work was completed and the reason it was completed with a smile. Without her and Leonor, our little penguin she has been coddling, there would be no reason at all.

There has been a number of persons who have been helpful beyond sanity. Tony Cornford, my supervisor, put up with my convoluted writing and provided keen insights that often led me to days of intense reflection. His capacity to think across issues amazes me.

Maha Shaikh has been a good friend to me and to this research. The debates with her have been some of the most challenging I have had. Her passion for the subject equals her passion for steaks. She has taught me to become a learned, ethical, and passionate researcher.

My parents, Nicole and Federico, have always been present. They have gone beyond themselves to be supportive. Their example of hard work is imprinted in my mind. My brother, Fabien, has been a staunch supporter for a long time. I remember with fondness when we would talk after school about what he had learnt.

My family in law—Paloma, Leandro, Carol, Richard and Mati—has been extraordinarily generous and kind to me. They have been an immense source of moral and encouragement. Richard's lessons about writing clearly will not be forgotten.

The department and the group have managed to create an inciting learning experience. Roser Pujadas and Nuno Oliveira's friendship has contributed particularly to making the learning process fun and intellectual. LSE's habitual challenging position to established notions has been a driver for my own perspectives on work. UCL's widely multi-disciplinarian approach to education and research has allowed me to pursue my interests in and beyond engineering.

Kostis Dryllerakis and Francicso García Morán, from the European Commission, have left me an unforgettable impression. Kostis took it upon himself to act as my mentor undeterred by my inexperience. His lessons on being a good professional and software engineer are invaluable. Francisco gave me his support and, despite his huge workload, he took time to listen and talk to an intern.

The CENATIC (the principal Spanish national center for the research and application of open source software), kindly invited me to work with them in their offices in Spain. We shared insightful discussions regarding the realities that had to be met by open source projects in Spain and beyond. Special thanks go to Manuel Velarde, Pop Ramsamy, Manuel Domínguez, and Jorge Martín.

The two institutions which gave me their financial support, the EPSRC for the beginning and the Fundación Ramón Areces towards the end, have allowed me to fully dedicate myself to this research.

Finally, the members of the openEHR project have been particularly forthcoming in their access. David's candour in his answers and his welcoming attitude were very helpful. All my interviewees stand out for their eagerness to respond to my questions. I would also like to thank all the participants in the mailing list, particularly David, Erik, Rong, Shinji, Thomas, Ian, Sam, Jussara, Tony, Seref, Pablo, Ed, Koray, Tim, and many others. After reading their interactions so many times, I thought I ended up knowing them personally.

Contents

List of Figures

List of Tables

Prologue

This thesis problematises the nature of requirements engineering, the part of software engineering which is responsible for the correct elicitation and modelling of a software's requirements. The identification and definition of requirements are often considered the first steps towards the building of software. The purpose of requirements is to define the functionality of the future software; it responds to the question: 'what should the software do?'. The activities that oversee the capture and definition of requirements have themselves become an important engineering activity inside the larger software engineering discipline, and have increasingly become guided by tools and modelling frameworks. Engineering requirements is not a simple task since requirements are often tacit to the users and complicated to communicate—thus hard to understand for engineers who are not intimate with the context. Also, the introduction of a new software is frequently perceived as a disruptive agent by the future users. In some ways, requirements engineering functions much like a Delphic oracle:

1. engineers are asked by the clients to build a new system (the context is framed);

2. engineers ask the users what they would like (identifying the requirements);

3. engineers model the users' responses (requirements are defined in a document which often forms the basis of a contractual agreement between the engineers and the clients. The future software's functions will have to satisfy the requirements document);

4. other engineers design, code and test the software.

Though sketchy, the previous points represent the main tasks in requirements engineering. Requirements engineering is much of a guessing game: engineers usually know little of the users' work. Nonetheless, they have to understand the specific needs of the users who often take their own work as natural and have difficulties to explain it. The requirements which are elicited and modelled from the users, are then coded into the software and introduced in the environment in which the software should function. Recurrently, the software does not meet the users' needs. This problem and its frequency is termed the 'requirements problem' in the literature. It seems almost comical that the hardest part of building a software system is, in fact, knowing what to build.

Upon examination, the literature tends to conclude that there is a disconnect between the elicited requirements, the users' expectations, and the reality in which the software is supposed to function. When the resulting software does not work according to how (for the users) it should work, users often resort to workarounds (which are socio-technical in nature). The root cause of this disconnect, it seems, is the messy nature of the world and the competing realities. How to capture the messy world? How to translate messy contexts into a machine which does not understand messiness? To solve this requirements problem, two main lines of investigation and practice emerged in the software engineering community.

The first relied on inter-disciplinary practices. It admitted the ambiguous, messy nature of the world, but worked close to the machine to simplify it. The simplification process would consider the world and try to understand it, but would streamline and order it along normative realities determined by the software and the requirements which define it. This line of investigation offered a positivist approach to the requirements problem and sought to take advantage of the disrupting effects of the software to improve and rearrange reality along (re-)engineered configurations. The idea was that the social world would be explored systematically with better tools to model it.

The second took an overtly social turn. The ‘soft’, social aspect became a methodology which would take the messiness of the world and create methods out of it. It took, essentially, a constructivist perspective: it tried to understand the interaction between the requirements and the software-to-be, and their effects across multiple interests and actors. Pictures that depicted the world would be drawn, shown to the users who would in turn draw their own picture. The user, consequently, would become more implicated in the process.

Though both positions took diverging paths—one more positivist, the other more constructivist—they shared the original insight that the world needed to be considered and its logic understood. Each position has its own merit: they proposed beneficial solutions and rethought, in their own way, the contexts in which requirements are generated and used. Nevertheless, both methods still saw requirements as stable entities with enduring, recognisable features. To this day, there still lacks a conceptual and—perhaps of greater importance—evolving conceptualisation of requirements. A requirement was and continues to be recognised as a functional, deterministic statement describing an aspect or purpose the machine needs to satisfy. It is this deterministic aspect that this thesis questions.

The enquiry is set in open source, a good laboratory in which to study requirements because it is subject to two contradictory affirmations, and is therefore controversial in nature. On the one hand, some authors believe it provides nothing new to the craft of software engineering. On the other hand, some suppose it is a paradigm shift, ready to challenge closed source methodologies, along with their organisational practices. Since open source development is still an open debate, there are no accepted views on its nature. Such volatility is more welcoming to the search for alternative views of consolidated concepts such as requirements.

The search for an alternative view takes, as a starting point, change instead of stability. Stability has been an accepted foundation, often undiscussed, despite being at the center of many ontological beliefs about requirements. Process studies, though not directly debating requirements, have contributed in questioning the problematic prevalence of stability as a normal, inescapable condition. Processual studies, instead, see change as being normal, almost routinised, perhaps even mundane, but essential in the functioning of organisations. Deleuze and Guattari's concepts, on which this study bases its theory, share this appreciation of the importance of perpetual movement (or the desire to). Starting from Deleuze's concept of rhizome, an ontological perspective in which requirements are in-becoming is proposed. Such dynamic and evolving requirements, if empirically supported, would imply different management necessities.

To look for such empirical evidence, openEHR is chosen as a single case study. openEHR is an open source health IT project, which, unlike other projects in open source, puts a large emphasis on the creation of knowledge artefacts. In contrast to many open source projects, openEHR is explicitly conscious about requirements, giving careful consideration to how they should be sourced and articulated. The principal source of evidence is the project's mailing lists, complemented with interviews. The use of these two sources helps to contextualise issues. Grounded theory methods are used to help the analysis of such a large quantity of data. From this a model emerges that relates identity, territory, and modes of engaging requirements. The emergent model is a meeting point between Deleuze and Guattari's concepts which are used to explain the empirical evidence.

The model is analysed, a chapter for each of the model's three principal themes. It shows requirements reach into the project's identity, the territories and market it tries to be a part of, and the way requirements should be articulated and how they shape realities. These realities, which openEHR tries to provide an answer to, are in movement forcing openEHR to move as well. But how should openEHR move? What requirements processes are better suited to these moving realities? Such questions—and how they should be answered—have deep consequences to openEHR, to the point of altering its identity and the places in which it seeks to provide solutions. To negotiate how requirements should shape realities, openEHR uses two principle modes of engagement: emergence and stabilisation. Though they respond to their own logic (their own 'mode'), the two are used in tandem by openEHR to articulate requirements in specific ways.

The analysis of the findings leads to the proposal of virtual requirements. That requirements are virtual does not mean they are not real, only that they hold an uncertain potential which can be actualised and realised in multiple ways. The distance between requirements' virtuality and their actualisation is not set; it can be negotiated by the different actors involved to source and articulate requirements in different ways. In this negotiation, time and space play an important role: if time is restricted too early, the potentiality of virtual requirements will not be as effectively explored, thus limiting their range of actualisation. Similarly, if space is understood too restrictively, the requirements' becoming will be rooted too firmly to certain framed territories and contexts. To benefit more fully from requirements, attention must be paid to create an environment in which they could come to express all their possibilities. It is important, then, for virtual requirements to be carefully considered as to how they come to express themselves in the assemblages of a project.

Part I

Stability and Change: Questioning a Paradox

Chapter 1

Introduction

> "The image is not singular, since it is a sedimentation of temporalities"[1]
>
> (Coulombe 2012)

Sitting on a bench, in a lecture room at UCL's computer science department, I hear an improbable paradox. It is one of the first lectures of my final year as an undergraduate. I do not expect a life-altering revelation when starting my final year reading computing engineering. But I receive one anyway: a majority of software problems can be directly traced back to incorrect requirements. Noiselessness in the class.[2] Apparently, such a simple question as "What should the programme do?" is really difficult to answer (Brooks 1995). The paradox is unsettling. "And we are told this in the fourth year?" I remember asking myself. Not really: in some way, that the users—that most external, irreverent, unexplainable meddler—could be problematic had been mentioned to us. Yourdon (1998), while citing Garry Weinberg, says "there are only three problems in developing software: people, people, and people." Snobbishly, I thought, the users were to blame. Not consciously guilty, but rather as a way of course, as representatives of those things that were difficult to understand: the ungraspable organisation; the political motives;

[1] My translation

[2] As Terry Pratchett notes, silence is not the opposite of noise, only the lack of it. The truer antonym is a strangely textured, almost muted, noise that is produced.

management; lack of general knowledge of computers; and all the other aspects software engineers have little control over.

The next years would serve as a continual reminder of the complexity of requirements. The small paradox multiplied. It went by different names: drifting, work-arounds, resistance to change, fear and domination, inscription and scripts. The dark side of computer science held many more accounts of requirements problems than the clean, crisp world of methods and tools. Everywhere, requirements provoked questions and debates. Everywhere except in open source it seemed. Did open source solve the requirements problem? Or did it just ignore it in the same way that Agile methodologies are sometimes accused of avoiding engineering requirements seriously? From there, questioning what requirements *are* is only a stone's throw away. These questions informs this thesis.

This chapter introduces the principal research questions that depart from that original paradox. The question: 'What is a requirement?' is refined by exploring some of the assumptions placed on requirements. It is in these (increasingly) disputed assumptions that a gap is revealed, from which follow a set of research questions and the aim of this work.

This chapter is structured as follows:

Inherited assumptions. Some of the assumptions prevalent in the field are briefly introduced.

Evolving contexts and the messy world. Requirements are briefly contextualised. This account describes how the concept of requirements has evolved over time.

Uprooting assumptions and research questions. This section presents the research questions.

Outline. The main message of chapters is outlined.

1.1 Inherited Assumptions

As with many aspects of computer science, requirements originated from the search for a solution to a given problem. The problem space—a term engineers use to define the space where requirements are to be found—was originally small. It included mathematical problems of ranging complexity, limited only to the architectural boundaries of computers of the time. Their resolution depended on their complexity and the time it took to go through the algorithmic solutions. Space and context mattered little, and meaning was unambiguous. However, as the uses of a computer moved beyond those of neatly defined needs such as those made for calculators—the computer's original use—so did the problem space. The solution to any given problem increasingly became relative to other possible, uncertain solutions. Computer science and software engineering tended to see the increasing complexity as a need for better modelling tools (Bell and Thayer 1976).

The difficulty was that computers did not do what they were supposed to because it was hard for people to know what they should do. Answering 'What should the system do?' became increasingly difficult as contexts became more complex and socially intricate. Beyond calculating ranges, matrices and other mathematical constructs, engineers realised that the world in which computers were moving was ambiguous, or at least had multiple solutions with relative optimal solutions. The initial reaction held responsible one source of the problem: the analysis of requirements. Requirements were not to blame, their codification was. To solve the requirements problem, what was needed was a healthy dose of engineering. By codifying requirements better, it made sense that they could be more precise and less ambiguous come the coding phases. By processing requirements to be more amenable to later stages of the software process, requirements were seen as an input to coders. It made sense given the philosophy behind software and requirements engineering: systems were supposed to be simple, predictable, unique, pre-existing. The machine part of the system was easily definable, limited to controlled and analysable inputs and outputs. The world, then, did not matter much beyond finding these requirements and their use as input for coding.

As computer usage grew, the purpose associated with requirements and their engineering changed (Jackson 1978). The focus shifted to eliciting existing phenomena of interest, instead of choosing the stable elements that could be easily described and coded. These phenomena would be stabilised by a systematic process from which requirements and machine specifications would be derived (Jackson 1995b; Jackson and Zave 1995; Sommerville 2005). The resulting requirements would be tied back to concrete stakeholder needs which would give a picture of the interaction between the machine (software) and the world (reality). This dichotomous arrangement of requirements—the machine on one side and the world in the other—opened the view for more elaborate models and approaches to the articulation of requirements. For example, goal oriented requirements engineering owes much of its ideas to that conception of the world (Lamsweerde 2009b). Certain aspects that had been largely disregarded took shape, for example, the means for the exploration of the problem domain where requirements would be found. Developing better ways to explore the world would lead to finding better requirements.

The act of considering the problem domain led to another reflection on the purpose of requirements engineering. Prior, only marginal thought had been put on the role requirements could take in actively exploring alternative solutions and look for the optimal one (Letier and Lamsweerde 2004). The problem domain, as a dichotomy between the world and the machine, opened engineering to contemplate the ambiguity of the world. Ambiguity was still an 'enemy' by necessity—something the machine (and possibly the world) needed to be purified from—but was, nonetheless, taken into the engineering process as a vital element to analyse for the machine to be built. To rein in the ambiguity of the complex world, models were invented that would piece complicated problems and allow easy analysis. Though the focus of the process remained the same—to cater for further activities along the software lifecycle—the domain was considered worth exploring. The philosophical assumptions behind requirements shifted from a predominantly positivist approach to software, to constructivist realism. The existence of competing models and processes ensured that, even though the problem domain was still believed as pre-existing the engineers' enquiry, care was needed in eliciting and articulating requirements. The wrong method could in certain contexts hinder the proper identification of the stakeholders' needs.

Competition between models helped launch a discussion, even if implicit, on the role and meaning of context in requirements engineering. Changes to organisational boundaries and the speed of change in the life of systems helped start divergent trends in research—more critical ones—with new ontological and epistemological positions (Bencomo et al. 2010). Open source has contributed in its own way to this as it is generally positioned against traditional requirements engineering because of the informal nature of requirements. It is for this reason that open source is a good laboratory to study the nature of requirements.

To research requirements in open source, it is necessary to problematise their conceptualisation and the different meanings they may take. This implies a move away from the assumptions of mainstream requirements engineering (and open source studies). Established requirements engineering views requirements as singular, stable objects that can be wholly identified from the world. For example, requirements are described as abstracted definitions (Lamsweerde 2009b). Open source, though indirectly questioning the nature of requirements, also holds assumptions of informal objects and processes where requirements are taken, seemingly, from the ether of collaboration. Even the theory of the commons, perhaps the only meta-theory that open source has, stabilises requirements by institutionalising them: they appear as objects waiting to be evaluated after a circular process of identification and validation (Benkler 2011). Requirements end up rationalised and subjected (objected) to stability.

1.2 Evolving Contexts and the Messy World

The aim of this thesis is to research requirements and to better understand their nature. Starting with the vague question 'what are requirements?', the enquiry quickly gains in specificity by limiting itself to the field of open source. Open source which has, as yet, seen limited research into requirements but indirectly questions their role by posing itself as an alternative to traditional requirements engineering. In open source, requirements have not been problematised yet. Not problematising them would lead to the unfortunate neglect of the opportunity to understand their constitutive nature in that particular domain.

Why are requirements in open source not such a debated topic? There are two reasons for this. First, research has deemed open source as an alternative method to traditional software engineering practices, but has, despite its attempt to distance itself, unfortunately inherited its language. Second, research has viewed requirements in open source as just another instance of requirements but placed in a specific context, without any particular attributes of their own—therefore not warranting any specific research. The first view has pushed open source into the world of Agile methodologies via its similar use of informal objects; it has thus avoided to problematise what open source—or agility—mean, and their consequences on requirements. The second view holds requirements to be universal objects, influenced in deterministic ways through the same mindful processes of purification. As a result, open source, as a domain for the study requirements, has drifted without gaining clarity. This thesis forms a response to this dichotomous view and proposes that requirements in open source, along with the processes that make and undo them, are constituted by and constitutive of their projects. Requirements are immersed, along with other objects, in the murky role of defining what things are.

1.3 Uprooting Assumptions and Research Questions

The strand of research which does not consider open source requirements as specific, is a derivation and an evolution of traditional requirements engineering. It is based on software engineering and its search for general problem-solving tools (Brooks 1995). To accommodate the growing complexity of information systems, the nascent field of requirements engineering saw that a rigorous and systematic approach was best to describe the needs of systems. This approach had the added benefit to be in a format that other software engineers, further down in the process of software production, would understand. This top-down, waterfall-like thinking was supposed to clean requirements from their messy origins and describe them through algorithmic statements. This kind of thinking, reminiscent of the natural philosophies, essentially proposed deterministic statements to translate worldly requirements into their machine-like equivalents. The messy nature of requirements was associated with the ambiguous human language and was not considered as an intrinsic property of finding what the program should do. This positivist view, although useful in the modelling of requirements for subsequent engineering phases, complicated the requirements' problem by over-objectifying requirements and abstracting the complexity away. But in every abstraction there is an extraction (Guattari 1995), and simplifying requirements also narrowed their purpose and role. The alternative that took a more constructivist perspective was more welcoming of the world's messiness, but maintained the similar consideration that requirements could be objects (albeit complex ones).

Searching for another understanding of requirements means staying clear of prevalent notions of stability which represents the principal assumption of established requirements engineering. Adopting the concepts of becoming, rhizomes, assemblages and potentiality from Deleuze and Guattari, this thesis operationalises them through the use of Latour and Callon's work on ANT. This study 'traces' and 'follows' requirements (Deleuze 2004; Latour 2005). This helps removing certain a priori conceptions of requirements and how they are not still, tamed objects, but potentially in perpetual movement. Doing so provides the opportunity to question what influence requirements have, and what influences them back. These concepts help answer four basic questions: *why* is a requirement the way it is, *when* is a requirement, *how* is a requirement developed, and *what* is a requirement in the context of open source. To respond to these questions, requirements need to be treated as just another element to study—the principal one—among the assemblages in formation: interests, open source, objectives, governance, values, but also temporal and spatial complexity, and other such actors usually put aside when studying requirements. This makes them indissociable to the processes that bear them. The use of this theory can help understand the way in which requirements evolve, are apprehended in the world, and the reach they have.

The choice of theory implies the use of a certain methodology. Deleuze and Latour, among other theorists, have influenced research protocols. They provide a theoretical position which can be useful to give voice to requirements. Empirical evidence is not formatted to fit a framework; on the contrary, the use of theory attempts to emancipate those actors and assemblages involved in requirements so that they may express themselves. Their views are not upheld as universal or true; but their assemblages and their evolution are taken into account in their own terms. This can be done by looking at perturbations, moments in which actors and assemblages question their place in the world. It is their disquiet that is to be revealed. By relating to the concerns of those that make up the realities in question, this type of theory grounds itself empirically. The data is gathered using grounded theory methods and their use of coding schemes.

The case study used here is openEHR, a health IT project that specifies clinical requirements for health systems. Health, as a domain, is an interesting field to study requirements in because of the multiplicity of meanings and complexity of the field (Mol 2008). openEHR is also an open source project, although unique in many aspects. openEHR is not the typical, infrastructural project with a predominant hacker culture. In contrast, openEHR has, for some time now, been discussing and coming to various understandings on the meaning of open source and how to handle requirements. Should they protect their requirements, controlling derivative products, or encourage their decentralisation? openEHR values requirements from a perspective other than solely an engineering one and can reveal aspects of requirements engineering that may be interesting.

The search for an alternative conception of requirements can be phrased in a list of research questions. The original question that motivated this research: 'what are requirements?', can be refined into several:

- What are the underlying assumptions underneath the mainstream conception of requirements? What are the consequences of these assumptions? Should they be questioned?

- Are requirements stable or tamed objects? Is there an alternative to the purification process engineering puts them through to make them useable? How are requirements sourced and articulated? Could (and should) there be an alternative understanding to the nature of requirements?

- What is specific—if anything—to open source requirements?

1.4 Outline

The thesis follows this structure:

Chapter 2. The literature review presents a chronological account of the different understandings of requirements and their processes. This chapter goes into detail about the assumptions underlining established requirements engineering and their evolution. Starting from one of the earliest acknowledgements of the requirements problem, several prominent engineering perspectives are reviewed: formal language descriptors, structured analysis, and the waterfall model. With the increasing complexity of work practices, another paradigm shook the foundations of knowledge on requirements: the 'world and machine' changed the relation between models and reality. Where, before, both were causal, the 'world and machine' view proposed the competition of various models and entities within a complex world. Requirements were not considered as natural objects anymore, but phenomena that needed to be explored within political contexts. Requirements engineering took, at the same time, a diverging route, one even more concerned with analysing the world and learning from it. Soft systems methodologies followed an approach which involved stakeholders as actors of change and introduced them into the development process. Along this line came a view that requirements were interactive: they performed the world as much as described it. To follow this line and study requirements as performative actors, the field of open source—and how it is helpful in departing from established concepts—is presented. This chapter opens the door for the theory to contend the accepted notions of stability that define them.

Chapter 3. This chapter presents the theoretical foundations with which to conduct the analysis and discussion. The theory used escapes mainstream understanding of stability and shows how change has redefined what is understood by stable. Thus, the theory links with processual studies which give a larger role for change to play. The main theoretical concepts are based on Deleuze and Guattari: rhizomes open the door to an understanding of requirements in-becoming with varying potentials and involved in complex assemblages.

Chapter 4. The method chapter first describes the reasons behind the search for a case study and the theoretical implications to the methodology. The theoretical considerations and the analysis of the literature guided the choice of openEHR as an in-depth, interpretive, exploratory case, principally because of openEHR's explicit consideration of requirements. The data collection choices are put in the context of the researcher entering the field and the gradual contextualisation of the project. The data analysis section presents the coding methods and schemes used. Though this research uses grounded theory methods, it does not build theory from the codes. However, an emergent model derived from coding acts as a meeting point for the theory to make sense of requirements. This chapter also presents the issues that arose during coding and justifies the decisions the choices made.

Chapter 5. openEHR and its context is presented and discussed. openEHR is a project that tries to develop complex clinical requirements that hold semantic meaning for their use in Electronic Health Records (EHRs). Unlike other open source projects, openEHR is not purely technical and is embedded in the complex domain of health IT. openEHR has a careful and explicit understanding of requirements and their role in EHRs. This chapter introduces the field of health IT and EHRs, briefly exposes the project's goals, and introduces some of the key concepts of openEHR.

Chapter 6. This is the first analysis chapter and is entangled to the other two analysis chapters (7 and 8). This chapter deals with openEHR's identity and its concern for *being* in the world. openEHR's identity has a direct influence on the way requirements are sourced and articulated. Three aspects of this relation between identity and requirements are explored: the type of collective that openEHR attracts and builds modifies the ways its requirements are handled; how requirements can reshape even foundational assumptions about the project; and the importance of openEHR's position in the market as a minoritarian force. This chapter shows the intimate relation requirements can have with the project's own identity.

Chapter 7. The second analysis chapter analyses the influence requirements have on (and receive from) the territories that openEHR wants to settle into. The way territories are contextualised and brought into openEHR denotes how requirements are sourced and articulated. Time features prominently in the issues openEHR faces when settling spaces: there is no referential time-frame which would restrict the exploration of worthwhile territories to explore. The immediate consequence is that the project does not have a cleanly cut visible frontier. The contextualisation of territories is difficult because the contexts themselves evolve and disrupt openEHR's own requirements and the way these are to be controlled. This chapter shows how openEHR maps the spaces it wants to settle and the issues it faces when tracing boundaries.

Chapter 8. The third analysis chapter analyses how openEHR engages requirements. Engaging requirements happens through two 'modes'—logics, as it were—which influence directly how requirements should be sourced and articulated. The two modes are emergence and stabilisation. The use of both modes is not contradictory; openEHR balances the advantages and disadvantages of both by balancing them out and valuing their contribution to requirements in line with its goals. This balancing exercise reveals the evolving, in-becoming nature of requirements in openEHR and how they take substance in the project and express changing needs.

Chapter 9. The discussion chapter first summarises the findings and their analysis into an emergent model which is given perspective through the use of Deleuze and Guattari's concepts. The emergent model shows requirements' reach through the assemblages they entangle to in openEHR. These assemblages are in-becoming, alternatively opening and closing the door to various potential realities that requirements shape. The high potentiality of requirements is termed virtuality and is the principal finding of this thesis. Virtual requirements are requirements which can shape realities in multiple ways. The shaping of these realities is observable when the requirements are being actualised and their forces enact realities. The virtuality of requirements is a consequence of their multiplicity and the competing realities that their actualisation can give rise to. Virtual requirements, if carefully managed, can be useful in accurately enacting realities in domains with high uncertainty.

Chapter 10. This chapter concludes the thesis, proposes a number of contributions, lists its limitations and summarises the main arguments.

Chapter 2

Literature Review

> "However, many operators sensed that the relationship between data and reality is a matter for inquiry, inherently problematic." (Zuboff 1989)

This chapter presents a straw-man of a literature review. It is so neither voluntarily nor involuntarily: it is impossible to account for 40 years of thoughts on requirements engineering without cutting corners. Some of these cut corners affect the chronology of the theme underlying this thesis: how have ontological beliefs about requirements evolved? Any clean divide between, for example, structured methods and the waterfall model belies the fact that both have been contemporaries, used together, and have shared the philosophical roots of top-down approaches. Pressman (1982), for example, treats them both as compatible approaches. However, structured analysis and the waterfall life-cycle can also be deemed contradictory: Chand and Yadav (1980), for example, argue in favour of a *rapprochement* between analysis and design and against Yourdon's own version of structured methods because they separate too neatly these two phases, thereby arguing against the neat boxing of development activities. Further, though past their prime, these techniques and paradigms, are still used in practice (Laplante and Neill 2004; Raccoon 1997; VanderLeest and Buter 2009), making any clean, chronological classification overly simplistic. Even some basic concepts like the waterfall model can be problematic and straw-men unto themselves: some argue that the waterfall model is often (or should be) used in varying degrees of conformity (Kruchten 2010).

The greatest difficulty, however, is that requirements cannot be discussed comfortably out of their context as if they were held inside a test tube, outside of time and space, and with clearly defined protocols to establish some truths about their matter. Though requirements have been, at times, bounded off and away from worldly preoccupations, requirements also represent evolving trends of various beliefs in science and the establishment of truth encompassed within closed artefacts (Kallinikos 2005). A review of requirements itself depends on other straw-man accounts of related issues, such as the waterfall model. The beliefs themselves are not necessarily epochal in a historical understanding (Popper 2002; Savage 2014), so reflections on requirements cannot be sectioned into neat, temporal packages. Maybe the ill-fitting categories that define the requirements straw-man make a useful point: requirements are a middle that, just like translation devices not only diffuse and hold knowledge (e.g. of the organisations' functioning), but also perform multiple realities, thereby enacting them along with their own selves (Ricœur 2006).

Thus, because requirements can be performative, the question 'Where does a requirement (or the belief thereof) start and end?' is irreducible. The literature review builds a straw-man out of necessity to attempt to explain the reasons why it is such. Despite of the rough chronology presented by this chapter, issues on requirements are by no means sequential. Rather, the chronology identifies certain ideas to certain time-periods, but does not conform one to the other. In other words, just as structured analysis methods still exists in a time dominated by object-oriented thinking (Champeaux et al. 1990), requirements and their ontological nature have been (and remain) controversial (Holmström and Sawyer 2010). Although these controversies are not always explicit, the problems revolving around the sourcing and articulation of requirements have been at the center of movements to change the way information system development should be thought of (Appan and Browne 2012; Davidson 2002).

Requirements' prominent place within the development of information systems compels an enquiry into their nature and the probable relation they have with the (seemingly recurring) software crisis (Curtis et al. 1988; Greenspan et al. 1994; Jason 1989; Lyytinen 1987). The so-called 'requirements problem' and its persistence has been empirically substantiated through a number of field studies showing that a large proportion of software defects were traceable directly to requirements (Group 1994, 2003; Juristo et al. 2002; Lubars et al. 1993; Potts and Catledge 1996). The requirements problem is the enduring causality between requirements and software defects (Lamsweerde 2000), and is an explicit acknowledgement of the requirements' prominent role in the larger concept of the 'software crisis' (Pressman 1982).

It is under this paradox—that there may be a lack of conceptualisation of requirements and that requirements play a central role in the definition of informations systems (Jarke et al. 2011)—that this straw-man-like review of the literature considers their nature. The chapter encompasses the nascent days of requirements engineering as a craft of its own when Bell and Thayer (1976) wondered whether they could be a problem. The red line running through the sections (summarised in figure 2.9) will try to show that a growing consideration and understanding of the role of the users in the determination of requirements marked the different perspectives on its nature (Pollock and Hyysalo 2014). The review starts in the late 1970's and early 1980's where the waterfall paradigm dominated (with a number of competitors such as pure evolutionary paradigms (Davis, Bersoff, et al. 1988), and a number of revisions that augmented its flexibility (Truex et al. 2000)); to risk-based, uncertain settings to which Boehm (1986) proposed his famous iterative methodology; to a growing recognition of the irreducible social environment, giving a larger space to contextual and performative complexity (e.g. soft systems). Finally, the chapter ends with a section presenting the characterisation of requirements in open source and how its differences to traditional conceptions can be fecund to enquiring over their nature.

2.1 The Early Articulation of the 'Requirements Problem'

The recognition of requirements as a problematic endeavour started in the late 1970's when the complexity of computer systems tried to follow suit with their increasing calculative power (Jackson 1983). One of the first articles to make explicit this troublesome relation reflects on the authors' experience with the rigorous articulation of requirements (Bell and Thayer 1976). A decade later, a survey of 17 large projects over their software design processes (that included requirements activities) attributes three problems to requirements specifically: thin domain knowledge by designers; changing requirements; and conflicts in their definitions (Curtis et al. 1988).

If the requirements problem has not been solved yet (Lamsweerde 2009b), its conception from Bell and Thayer's (1976), to Curtis's et al. (1988), evolved alongside the parent field of software engineering to expressively take into account the organisational environment as key to successfully describe requirements. The 1980's are marked by change in the software development paradigm because of the growing recognition that information systems, in their totality, are irreducible to a machine; the world is too large for any machine. The waterfall model shares the top-down approach of structured analysis, and because of its inflexible nature, helped further the simplification requirements, subsequently detached them from worldly concerns.

In this section, the earliest understanding of requirements is presented. Their definition and processes fit into an era marked by a widespread acceptance of the waterfall approach to software engineering (Royce 1970), and an emphasis on information flow modelling (Coad and Yourdon 1990).

2.1.1 Early Requirements Representations

The basic premise of requirements engineering, at that time, was to understand what a system was to do (Jackson 1978). At the most basic, software requirements are descriptions of "functions that the software must perform, but not how they must be performed" (Bell and Thayer 1976), or "all aspects of system development prior to actual system design" (Ross and Schoman 1976). There are two points to take out from these definitions: the first point is a view of the software system as a functionally described machine which takes inputs and computes outputs, the second point evidences the nascent field of requirements engineering which should hold different concerns than the closer-to-code design phase. Bringing these two points together sums up the purpose of requirements engineering: describing all of the machine's functions before any implementation activities.

In the 1970's, when requirements started featuring prominently in general software engineering literature, formalistic statements were the state of the art; these are highly structured statements describing requirements as functions over inputs and outputs. Increasing the rigour of a requirement's description, it was hoped, would increase its veracity. Bell et al. (1977) proposed such a formalistic language which would rigorously provide definitions of a machine's computational steps (c.f. figure 2.1). This is in opposition to unstructured requirements and informal definitions which could be misleading (c.f. figure 2.2). The formal definitions, instead, were to provide unambiguous descriptions to automate much of the engineer's work to help him "use his creativity, [thus] the system should be natural and flexible; it should allow him to express requirements in terms of concepts which are familiar to him" (Bell, Bixler, et al. 1977). The emphasis is on the engineer's use of concepts "familiar to him", not those used by the machine's future stakeholders.

Beyond the natural and formal definitions of requirements, structured analysis methods try to add a more holistic perspective to the relation between requirements and the world. This is done through the explicit diagrammatic representation of information flow (i.e. information transformed by functions) through systems (Chapin 1978; Mylopoulos et al. 1999). The principles of structured analysis are the same as those underlying natural and formal representations of requirements; both deal essentially with the modelling of flows through the definition of the trio inputs-transformation-outputs. In structured analysis, this trio has various levels derived in from the top down: from highly abstract and conceptual views of the whole system, to lower levels representing the system-as-solution (Chang et al. 2001). The systematic, functional, top-down derivation of a requirement is called *functional decomposition* (Coad and Yourdon 1990). Functional decomposition does not refer to an increasingly precise understanding of what the system should do on the whole; instead, it gradually details the routines and sub-routines that the machine should execute.

Compared to formal language requirements, structured analysis might not seem much different. For example, both Bell and Thayer (1976) and Chand and Yadav (1980) see requirements as information that is formatted in some unambiguous way. Both see in the nature of requirements a precise rendition of a machine's working routines. The difference is a clearer relation between the functions and their effects on information through a diagrammatic representation. The drawing of various levels of the system-as-solution domain was intended to provide a hierarchical ordering of data transformation (hence a structured analysis). For example, figure 2.3 shows a data flow diagram at a certain level of abstraction (Peters 1980). Data flow diagrams, as an example of structured analysis techniques (Chapin 1978), provide "the software engineer with guidance on how to define the software given that he has a model of the data flow which will eventually occur in the system" (Peters 1980). The system, thus, *is* the machine; requirements are algorithmic information describing the inter-related routines of a machine. The possibility to diagrammatically see functions in action helped to see the "forest for the trees" (Pressman 1982, p.97) and gather an overall picture.

Despite being machine-solution oriented, structured analysis, with its focus on information flows and their transformations, provides an exploration mechanism with which to involve the user (Yourdon 1977), despite a loss of meaning in the lower levels of the data flows (Coad and Yourdon 1990; Sneed 1989). The ability to tie functions together and the necessity to understand the expectations behind them—and their larger consequences—pushed for the involvement of the stakeholders more readily (Ross 1977). The implication of the user is an important idea that will gain more prominence throughout the evolution of requirements and software engineering (Iivari and Iivari 2011; Jackson 2008).

Notwithstanding the progress made by structured analysis in the philosophical understanding of requirements and their role as transformative information that is not so easily plucked out of the ether of contextual reality, the 'requirements problem' remained. Partly to blame is the waterfall approach which reinforced the reliance on top-down, solution-oriented approach of structural analysis. The waterfall glued requirements as a well-defined activity which would output a clear and stable deliverable. The waterfall model, though part of the general software engineering work on software life-cycles, helps explain the expectations placed behind requirements in the overall engineering effort.

2.1.2 The Waterfall Simplicity

The waterfall model is a top-down approach and a contemporary of structured analysis (Raccoon 1997; Sneed 1989). The model consists of a series of activities carried out in sequence and in which backtracking is discouraged (Wolff 1989). This is not always the case, and the early promoters of the waterfall did not specifically deter from returning upwards to previous activities (Davis and Sitaram 1994). More than a theoretical imposition, the recurring problems of backtracking that the waterfall suffered from might rather be attributable to the legitimisation of the deliverables at every phase. If taken to extremes, analysts would be confronted every so often with a "go/no-go decision" (Cave and Salisburi 1978), forcing them to make decisions that might be invalidated in later phases despite the looming costs of late corrections (Boehm 1984b).

It is this funnelling force to which requirements are subjected that proves of interest here. Requirements are first specified in one of the earliest stages, when there is little yet known about the problem domain (c.f. figure 2.4). The requirements deliverables, after being verified, are then sent to the designers who turn them into functions ready to be implemented. This passing off admits, at least implicitly, that it is possible to have sufficiently complete requirements early on

(Buss 1981), which is usually not the case (Rowen 1990). For example, Heninger (1980, p.2) recounts the difficulty of creating a complete specification of an existing aircraft prototype “in spite of the availability of a working program and experienced maintenance personnel”. She goes further: “None of the available documents were entirely accurate; no single person knew the answers to all our questions; some questions were answered differently by different people; and some questions could not be answered without experimentation with the existing system.”

The difficulty encountered by Heninger and her team to understand requirements, let alone provide a complete specification, resides in their multiplicity: they hold different meanings to different people. The objective of the waterfall tries to lop off the ensuing uncertainty of meaning by putting a large effort in understanding the problem domain early. It does this by stabilising requirements by freezing or rooting them in a document that constitutes a homogeneous representation of meaning (Cusumano and Selby 1997; Lyytinen 1987).

The problem is that this freezing ultimately drives and coordinates software engineering efforts though objectifications (Scarbrough et al. 2012), which are, however, inflexible to change. Falsely rooting requirements down and artificially removing their multiplicity consequently affects their representative capacity. Some authors consider requirements as boundary objects with interpretive flexibility (Barrett and Oborn 2010; Doolin and McLeod 2012); if this flexibility is removed, the world portrayed is at best inaccurate, and at worst difficult to alter. Some views become invisible, possibly discarded, while others can be given an unmerited prominence (Davidson 2002).

From the onset of the waterfall, therefore, requirements are narrowed in scope instead of being expanded (Jackson 1978). There is little time and little space after the initial steps given for the search of participant identities, despite its crucial role to understand the complexities of a given situation (Callon, Lascoumes, et al. 2011). The rooted and narrowed requirements gain an unbecoming legitimacy having been abstracted by expert analysts and implemented by developers taking the mythical role of ‘high priests’ (Urquhart 2001). They ‘unbecome’ in the sense that they evolve little, their potentiality cut early (Deleuze and Guattari 1987); the requirement might be well described, but a poor representative of needs.

The experts charged with the burden to make these decisions serve a similar role to that of the scientist-philosopher depicted by Latour. Using Plato’s cave as an allegory, the expert removes himself from the worldly pollutants that could alter his access to truth (Latour 2004). In the waterfall period, politics are such a pollutant and the objective of the expert was to remove or to limit its effects (Hirschheim and Klein 1989).[1] The expert analyst abstracts requirements out of the messiness of life and standardises them, attempting to produce a universal value useful along every possible user of the system (Ciborra 2002b).

Law (2002, p.150) echoes this standardisation when he speaks of the disappearing of the political, or “that which cannot be said, or at any rate cannot be said in the right place, removes itself from the Political becoming something quite Other.” Once the requirements document has been written, it is difficult to nudge it through small politics; requirements having come to represent the universalised big Politics. Local situations, small voices, Others, they are all equally crushed beneath the enlightening power of technical requirements that have been purified from the organisational politics by scientific processes.

This is, at the same time, the strength and weakness of formalistic and structured requirements: they assume that the software system can be strongly anchored to truthful requirements, and because they are truthful, their place in the world should be unproblematic. As Crotty (1998, p.27) says of positivism: “science ascribes no meanings at all. Instead, it *discovers* meaning, for it is able to grasp objective meaning, that is, meaning already inherent in the objects it considers.” The requirements processes, thus, are more about measuring the world than producing it since the measurement relates objective, pre-existing, and essential truth values (Law 2004). Requirements thus also act as a social instrument of acceptable

[1] This objectivist view, while still holding some truth in that “the criterion of their [abstractions] validity is an objective correspondence to material reality” (Jackson 2012, p.496), is much less extreme in contemporary requirements engineering in which the user’s complexity is tackled explicitly (Nuseibeh and Easterbrook 2000). Thus, objective truths are revealed, not through the removal of the expert from political ambiguities, but through his participation.

change (Munson 1981), at the expense, under the waterfall, of the performativity of multiple becomings.

2.2 Exploring the World to describe the Machine

The top-down approaches of structured analysis and waterfall modelling gave way to more flexible paradigms such as object-oriented analysis, Boehm's (1986) risk-oriented, spiral method, and Jackson's (1995b) view of requirements as world and machine phenomena. This section introduces this evolution in the understanding of requirements as a more flexible object which represents reality is less clear-cut ways, requiring an explicit and repetitive exploration of the problem domain. The methods of analysis and the overall spiral methodology (indicating a departure from the waterfall's usage of requirements), are presented through underlying similarities. First, requirements are shown to be perceived as different kinds of objects, being much closer to the problem domain than the machine-solution of top-down approaches. Second, the overall conception of the machine as a system has shifted; it is no longer considered as a clean separation between inside and outside, but as a complex transformer of inputs.

2.2.1 A Different Kind of Objectifying

Instead of equating requirements with a system's functions, requirements gradually came to be seen as materially diffused. Starting with Jackson, the correspondence between the model and its accurate representation of reality was discussed. Instead of seeing requirements as functional decompositions of a machine-solution, Jackson argues that a system should be viewed as "a model of reality with which it is concerned" (Jackson 1978). The consequence of this means that requirements are mediators; their essence is not functionally evident because it does not translate directly into machine functions and sub-routines. Instead, "requirements are about the phenomena of the application domain [the world], not about the machine. To describe them exactly, we describe the required relationships among the phenomena of the problem context" (Jackson 1995a, p.169).

The wording is telling: requirements are described *exactly* when they describe the *required* behaviour between the world and the machine, and requirements reside in the world[2]. The world and the machine, when portrayed as intersecting Venn diagrams (see figure 2.5), show requirements as interface-beings. As an interface, requirements can (objectively) mediate the world in which they reside (they tell of objective realities that happen), and describe (or should do so) the machine which interacts with the world.

Treating requirements as interfaces casts doubt on the strong positivism behind their nature; after all, there are many possible models of requirements to accurately represent reality, yet some are better than others (Letier and Lamsweerde 2004). If there is still one reality (or world) in which one machine is a part of, it is indicative that Jackson's understanding of requirements as models of phenomena is still positivist, only less so because their modelling admits the possibility of misrepresenting reality (of which there can be many, possibly conflicting models). The work of the engineer, then, becomes one of sorting through the possible modelling. The direct consequence is a shift from the machine-solution oriented engineering that was more common in the top-down approaches, to one focused on the problem domain and closer to the user and stakeholders.

To see requirements as interfacing puts the world back into the limelight because the world is mediated (and re-mediated) through the machine (Kallinikos 2011). As an interface, requirements are in a state of "being on the boundary. It is that moment where one significant material is understood as distinct from another significant material. In other words, an interface is not a thing, an interface is always an effect" (Galloway 2012, p.33). There is a differentiation taking place, of which requirements are in part the describers, and in part the performers (or rather, to describe is to perform). Perhaps this is the reason why the world and machine metaphor has gained such popularity to explain the place of requirements

[2]Although requirements are deemed by some authors to belong to the world (Lamsweerde 2009a), such a boundary can be considered artificial: This is especially the case if requirements are treated as *milieux* or *plateaux* (Deleuze and Guattari 2004), or even interfaces (Galloway 2012)

in software engineering (Berry 2011): it provided an engineering lens to account for the rapidly evolving meaning of the requirements in the world.

This is in contrast to functional decomposition: if a requirement needs to be validated by the client, then it is more than just a programme's function or a number of agglomerated sub-routines (even though it would be described diagrammatically through data-flows). For Jackson (1983), the function and its decomposition into lower-level sub-routines is not the main concern. Instead, he argues that the starting point for requirements should be the synthesis of the problem domain and the hypothetical interactions of the machine, without which there would be no context when requirements are specified (Jackson 1983). A requirement, thus, sits somewhat uncomfortably between the social world from where comes its context, and the machine, which offers a number of ways to implement the requirement.

The world and the machine, when portrayed as intersecting Venn diagrams (see figure 2.5), shows the difficulty of pure top-down approaches to black-box a software's requirements as a conglomerate of machine-functions. Kallinikos (2006), building on Luhmann, has termed this aspect the 'closure' of a technology's functionality. Closing off functions can be done in different ways. In structured analysis, the closing happens when functions are decomposed successively, gaining further legitimacy at every lower-level of analysis; any change will have to confront with the number of derivations that went through defining the function (in this sense, the derivations would be repetitions of the same since they would be causally related (Deleuze 2004)). In the world and the machine paradigm, it is much more difficult to isolate the functionality from the world since the machine's influence on the world is subjected to comprehending the problem domain, and thus, the requirements.

2.2.2 Requirements as Phenomena

If both structured analysis, and later, Jackson's and Boehm's conceptions of requirements serve the same purpose, namely, to objectify requirements by selecting their functionality and objectifying them, do they differ at all? The difference is related to both the nature of requirements (what are they in the world?), and how they should be approached (how can requirements be described?). By treating requirements as transformable information, structured analysis makes the machine the sole responsible of re-ordering the world. Jackson, instead, treats requirements as world phenomena (c.f. figure 2.5) on which the machine acts upon. The ensuing system is less cleanly definable and prone to the vagaries of the social world.

Jackson (1995a, p.143) defines phenomena to be "what *appears* to exist, to be present, or to be the case, when you observe the world or some part of it in a domain". On one hand, there are world phenomena, from where requirements are found; on the other hand, there are machine phenomena—machine-specific requirements the machine alone is responsible to see satisfied. Shared phenomena are requirements that are implemented, or should be implemented in the machine, but which also depend on world phenomena to be satisfied (such as a user pushing a button). In this sense, these shared requirements bind together the world and the machine. Although both are distinguishable analytically, the machine's purpose and incidence on the world is made clearer and more complex.

The nature of statements is different in the world phenomena than it is in the machine phenomena. Ontologically speaking, the interesting requirements (for the engineering of the system) are those that exist in the intersection between the world and the machine, for example, whether they are describable (through translations) to machine phenomena. Analytically, though, these two worlds are separate, conforming to different rules and adapting to contexts. The world phenomena which have no direct incidence on the machine (but may have an indirect one) are explored and generally assumed, unless they are counted as a possible risk.

This machine-world perspective widens the scope away from information flows, data and levels of systems, to actors, phenomena and "happenings" (Ross and Schoman 1976); what is potentially interesting to requirements has increased in size. The context is present and is explored: to understand what is a shared phenomena, world phenomena have to be investigated. The existence of the (single) reality is still assumed, but the variety of models and alternatives open requirements engineering up to wonder about alternative problems and solutions (Duboc et al. 2013). The role and prominence of the machine has also shifted; it is ontologically inseparable from the world (Jackson and Zave 1995). The shared part of the world, the boundary, is much more malleable than pure data flows through well-defined systems.

That requirements are part of the world and act as interfaces between themselves (e.g. as description) and reality(-ies) indicates a loss of control away from the engineer. Top-down approaches, as has been criticised, tried to deal with this problem by freezing interactions and simplifying requirements as machine-functions. The loss of control directly influences the nature of requirements because they can no longer be considered 'wholes' or complete, as they would in the previous, more positivist accounts. Instead, requirements, as a body of functions become superseded by a number of partial concepts: viewpoints (Nuseibeh, Kramer, et al. 2003), for example, represent a stakeholder's partial understanding of the system that need to be accommodated (Boehm 1998); or, experiential phenomena that indicate some transformative movement (Jackson 1995a); or, even, constructed narratives (Barrett and Oborn 2010; Urquhart 2001).

Though the terminologies are different and denote distinct aspects of requirements, it is the erosion of requirements as a complete thing represented as the trio input-function-output that is of note. How should incomplete requirements be approached?

2.2.3 Approaching Requirements: Risks, Uncertainties, and Imbrications

Considering the need to model reality is indicative of the uncertainty behind the process and the representative power of requirements. It puts some distance between the requirement-as-function and reality; a requirement's description needs to be mediated through a model. As Gause and Weinberg (1989, p.9) recounted a saying common in the Swedish army: "When the map and the territory don't agree, always believe the territory." Boehm's (1984a) instigations to reflect on both the validity (useful system) and veracity (correct system) of a software echo the controversial multiplicity of requirements. These are not only difficult to see implemented correctly; their interaction with world actors needs to be validated as well, making this process an uncertain one.

Boehm's (1986) spiral model is a testament to the uncertain and risky nature of the world and is a cornerstone for the inclusion of requirements into contemporary software engineering paradigms based on iterative and incremental approaches (Lamsweerde 2009a; Mathiassen, Saarinen, et al. 2007). The spiral model admits implicitly that complete or perfect requirements are not likely to be found easily (c.f. figure 2.6 representing the multiple and repeated tasks of validation and verification). 'Perfection' can only be approached through multiple iterations, incrementing step by step the deliverables (requirements document and others). This approach echoes Parnas and Clement's (1986) words on rational processes and why they should be 'faked'. Under such a view any requirement might not be *the* complete requirement; what matters is getting as close to it as possible so that it could be considered as such. The reason is the number of iterations which give confidence to the degree of accuracy in depicting reality. In other words, requirements might be marginally imperfect, but their representation of the single, extant reality is taken to be a symbolic truth, becoming legitimised through the iterations and increments as almost perfect translators of reality. Such objects invariably hold an increasing objective force the more increments are carried out.

This objectifying process is shown in figure 2.7, which is a transversal cut of Boehm's spiral model. This view puts in evidence that the spiral model is an iterated version of the waterfall model and reinforces the existing set of requirements by incrementally building on them. Each iteration is only possible through the incremental accretion of deliverables, passing knowledge from one iteration to the other, until the 'right' deliverables are developed, whether code, requirements documents, or test suites. All increments are driven towards the satisfaction of the initial stated goals and requirements in the first increments, which tends to limit the possibility of deviation.

Objectifying requirements in such a way is problematic: are the repetitions (or iterations) not creating distinct requirements? Or do they serve the purpose of idealising requirements so that they lead future iterations? In other words, are the iterations repetitions (where difference is admitted and even contingent to repetitions (Deleuze 2004)), or are the same requirements reproduced? In the spiral model, this question amounts to asking whether phenomena are subordinate to their representation as documented deliverables.

Following Kallinikoss (2006) concepts, technology is necessarily about the closing of operations. Under this view, technological systems are designed to be closed around selected, unambiguous operations.[3] When Kallinikos (2006, p.37) writes: "Unforeseen relations cannot be handled *in situ*. They could possibly be incorporated into the model by the programmer in a future periodic revision of the program…", he points to the temporal contradictions between the world and the machine. Though the machine is charged with intentional change on the world, these intentions need to be revised once the machine is deployed. This disjunct feedback loop is reminiscent of the historical division between the complex social world, and the unambiguous intentions of the machine (the Taylorist scientific management in particular (Zuboff 1989)).

To solve this contradiction, Boehm's spiral model seems to ally well with the world and machine paradigm: both ultimately attempt to select those requirements by experiencing the world, by iterating dips into the world. The same does Jackson (1995a, p.143) when he describes phenomena as those things which appear and disappear through the use of language, adding that "Each method supports, encourages, or enforces a particular way of seeing the world." This is not far from Derrida's (1967) point when he wonders why originality is given to the oral word and deprived from the written form as if the spoken word came necessarily first; as if the written form (the model) could not be eminently influential on the semantic

[3]Part of this is being reviewed by agent-oriented requirements engineering, a field which studies a system's adaptation and re-articulation of its requirements contingent to environmental stimuli (Bencomo et al. 2010; Chopra and Singh 2012).

foundation of the spoken idea's substance (the world).

That the representation of a requirement could narrow the experience of stakeholders and the understanding of the problem domain is important. Such narrowing means that it is not only the unambiguity of the computer's binary manipulation that affects requirements, it is also the method used that is responsible in the 'cutting' of a reality, making apparatuses to see the truth from the false (Barad 2007). This 'cutting' of the world and the machine takes place at various steps. In the machine and world paradigm, the first such 'cut' is to separate the machine from the world, somewhat cleanly, as if one would be imbricated into the other (Leonardi et al. 2013). In the spiral model, it is separating and focusing on activities between the requirement's side and the implementation side (Céret et al. 2013), despite the concurrency of software engineering processes (Davis and Sitaram 1994).

Behind the separation between the world and the machine hides a discussion of systems: Jackson's and Boehm's greatest merit is, perhaps, to have shown how much one was permeable to the other. Instead of brushing off and simplifying early the complex interactions of the social world, the paradigms proposed by both authors sought out an unity of the machine-system that was to be achieved through the differentiating possibilities it could bring onto the world. [4] Luhmann (2013, p.63), when defining systems, mentions particularly this paradox between the unity of the system and how it differentiates: "the important issues consists in the fact that the system draws its own boundaries by means of its own operations, that it thereby distinguishes itself from its environment, and that only then, and in this manner can it be observed as a system." In this sense, "a system is not a unity, but rather a difference, and, as a consequence, we have to face the difficulty of conceiving the unity of the difference."

Deleuze (2004) accords a whole book to *difference* as a concept onto itself, the "difference of the difference" (Zourabichvili 2012, p.103), free from the false difference that is always rooted onto an originator; a source which imposes its

[4]Along a general movement that included object oriented analysis.

own moral value system thereby, reproducing itself. The top-down approaches did precisely that very early on: they 'froze' requirements (both the idea of a requirement and its representation) (Sneed 1989). Consequently, requirements risked being reproduced from early conceptions. Jackson's and Boehm's ideas translated to an unsettled boundary, one which itself needed to be argued. Where should the machine stand? Is it inside the world and the interactions are more messy, or is it outside, with a clean interface between the two? Part of the answer resides in the distance between its representations (not being cast into machine-representations of functions), and the present's capacity to enact and interpret both its dimensions of past and future (Hernes 2014; Zourabichvili 2012). Thus, the 'cuts' in themselves are not necessarily negative; they translate a necessity of translation to machine code. It is important, however, to consider how this cutting is done because it influences not ""what is being?" but *who*—or how—is being?" (Zourabichvili 2012). "Who draws this distinction..." between the system and its environment which "is the difference that constitutes a system" (Luhmann 2013, p.63)?

2.3 The Socio-Technical Gap: Requirements as Interactive Objects

Questioning the imbrication between the world and the machine, that perhaps the interface—however multiple—is not so cleanly delineated, tantalises the concept of requirements. If considered as interfaces, and given their possible multiplicity, it becomes much more difficult to understand when requirements begin, why they are needed then, and who is involved in the process and at what point in time they are involved. As Finkelstein (2012) notes: "We have tended to view requirements as a discrete task in which we engage with the customer (a sort of shorthand for stakeholders) on an occasional basis. [...] You could say the software is a manifestation of the relationship achieved through continuous interaction and immersion in the business." Finkelstein's words could be interpreted in the following way: requirements are interfaces (a manifest relationship, one which mediates), but as interfaces, they depend on the level of interaction to be described (how much they mediate). It is this interaction, and its intensity, that is the purpose of discussion in this section.

This section deals with a 'softer' understanding of requirements, where it is admitted (at least implicitly) that requirements are irreducible and performative: the 'construction' of a requirement leads to the construction of the world-machine, and vice-versa. Because of the irreducibility of the social world into the technical world (Ackerman 2000)—its bridging contingent to advancements in artificial intelligence (Dreyfus 2007)—, there is some sense of futility behind constructing systems. As Checkland (1985a, p.821) says: "In Churchman's [(1971)] language, S.S.M. [Soft System Methodology] is a "Singerian Inquirer", one which accepts that inquiry is never ending and is intent, in "an heroic mood", on both attacking and defending the *status quo*." Compared to the previous sections, this one considers the possibility of defining optimal requirements as elusive: just like Zeno's paradox, the more requirements represent, the more of reality they create, creating in turn the need to represent some more, and so on.

This section looks deeper into the irreducibility of requirements and discusses why treating them as interactive entities is helpful in teasing out their nature. The interactive aspect of requirements represents a third way to understanding the nature of requirements. If the two previous sections related requirements and their representation of reality, the third way questions this relation, thus relying more on interpretive processes that create and debate the construction of reality (if indeed there is one). The result is a *rapprochement* between the world and the machine, making one indistinct to the other (even for analytical purposes). The operational closure enacted by the machine, implicitly accepting the separation of logics between the world and the machine, is thus put to the question.

2.3.1 Soft Systems and the Interactive Requirements

Soft systems is more a heuristic methodology than a prescriptive method (Checkland 1985b). It contains four different kinds of activity that are not necessarily sequential, but can, in the most simple of cases, follow this pattern (c.f. figure 2.8): initial perceptions of the situation are explored; models are made that describe some actions; the models are related to the "real" situation; and, finally, an action is taken which supports desirable and feasible change (Checkland and Poulter 2006, p.13).[5] The perceptions behind a situation are indicative of the problematic locality which needs to be explored by the stakeholders.

At first glance, there is much to connect Checkland's soft systems methodology to contemporary efforts in requirements engineering, and specifically, Jackson's world and machine paradigm, with some authors arguing for their conjunct use (Ulrich 2012; Water et al. n.d.). Just as Jackson talked about phenomena and the exploration of the problem domain, Checkland speaks of modelling relevant 'happenings'. Such 'happenings', once modelled, form part of points of views (specifically, the *Weltanschauungen*, parts of the 'root' definitions that describe the world-views). Such a *Weltanschauung* gives meaning to the root definition; it puts the world itself—and its arrangement—in a problematic and possibly conflicting position which needs to be explored. For example, the perspective "the Olympic games from the perspective of the host city" organises meaning around the host city (Checkland and Poulter 2006, p.38). The world-views and root definitions could be seen to resemble problem-frames in which goals and requirements operate, a sort of "problem class" that defines the world (Nuseibeh 2010, p.346).

The difference comes when the multiplicity of the *Weltanschauung* is discussed: "It does not bother to ask which models are 'correct', since the models are not

[5] It is, perhaps, contradictory that Checkland and Poulter (2006) talk about the "real" situation as if it could be known at all, especially since they put such an emphasis in the impossibility of completely learning everything about the situation. There are various ways to clamber out of this predicament: one is to adopt a structurationist ontology where "real" refers to unmoveable constraints or structures that influence situated action (Giddens 1984). Another is to adopt a more processual understanding where "real" is not *the* reality, but becomes a force created by change itself through the exploration of the domain. Such an exploration actualises the interactions and objectives of the systems, which are thus interpreted differently by being confronted to various possibilities.

would-be descriptions of part of the world, they are accounts of some ways (among a multiplicity of ways) of perceiving it." (Checkland 1985a, p.822). The representation of the world and the world itself are unhinged: the models have become accounts that are deliberately removed from specifically enacting it (Mingers and White 2010). In the world and the machine paradigm, the requirements are clearly represented as interactive shared phenomena: the general goals and objectives found in the world would be directly or indirectly operationalised by the machine (Jackson and Zave 1995). This is specifically rejected by the soft systems methodology because the world and the objectives therein are too complex to be easily delineated into clean, singular systems (Checkland 2000).

Checkland's point about the perception of the world is important to further differentiate the third way: the singular system does not exist, except in the most simple cases (Checkland 2000). In norm, however, systems are complex systems which themselves are interpretable in a number of ways (Checkland and Poulter 2006). The interpretivity of systems means that the same system could be modelled and still be understood differently. In this sense, the systems as wholes and parts are not unique and essential as some believe (e.g. (Robinson and Wilson 2003)), they do not translate *a* totality, but one of many possible and feasible systems. Work in STS has given a particular emphasis on the irreducibility of concepts, however standard (Bowker and Star 2000), however technical and scientific (Mol 2012; Mol 2002), however popular (Star 1990), however engineering-like (Law 2002), however quantitative (Beunza and Stark 2012).

The recognition of this ontological multiplicity—that the same things are interpretable in a number of ways—forces SSM to reconsider the reality of systems and the capacity of analysts to imbue them with objective statements (Ulrich 2012): instead, problems need to be unfolded. To unfold problems, either the analysts pose as experts and mediate changes from the stakeholders (as would requirements engineering tradition, e.g. (Robertson and Robertson 2003)), or stakeholders learn the the methodology and become participants of change, blurring the lines that separate him from the analyst (Checkland 1985b; Ison et al. 2014). This is an often overlooked aspect of soft systems: that depicting the *purposeful action* of stakeholders correlates to knowing, or rather learning, this purpose. The stakeholders are purposeful actors who enact systems transiently, depending on their moving understanding, and which properties emerge in time (Checkland and Poulter 2006).

It is this closeness to the stakeholders—beyond the simplified categorisation of formality versus informality which pits 'hard' against 'soft' systems—that effectively breaks part of the concept of requirements as an interface. The closeness between expert, analyst, and stakeholder shatters a 'fourth wall', as it were, propelling a different narrative to explore the problem domain. The teasing out of requirements becomes an interactive matter, one in which the stakeholder is no longer queued in by the analyst (Urquhart 1999), but a knowledgeable and capable peer. Ackerman's (2000) socio-technical gap is bridged differently, influencing the ways requirements are conceived and represented.

2.3.2 Requirements and its Multiple Bodies

The interactive multiplicity of requirements is a prickly problem: if a requirement can be represented in many ways, then *what* is a requirement? Is the representation of a requirement equivalent to the requirement, or, instead of being a faithful photograph, is it a painting approaching the 'fundamental' requirement through method? The previous sections have shown an increasing uncertainty towards a clean definition. The (elusive) answers to this 'what' question have shown the multiplicity behind the concept of requirements and the increasing relation to a problem domain have brought it to the sphere of the socially constructed (Hacking 1999).

If requirements are constructed—that is: they do not exist independently in the natural world—requirements are shaped by how they are approached (a key tenet in soft systems) and how they are valued. Their construction influences this valuation, as Preda (2007) argues: "Only by making value calculable can market actors compare (fundamental) value with the price, and decide whether securities are over- or undervalued." While Preda's work analyses the evaluation of securities through devices—a world apart from requirements engineering—the similarities are revealing. Structured analysis looks for those requirements that describe the system's functions from the abstract to the machine code. Soft systems also believe in a fundamental value, but one which is made. Similarly, Hertzum (2004) speaks of an uneasy balancing by engineers between 'hard', contractual requirements, and a never-ending, open-ended exploration of the problem domain. The tying down also contributes, enables even, their search.

It is telling that some requirements need to be tied down to allow for the exploration of others to get to their own stable definition. Perhaps this stabilisation is due to the increasingly dynamic nature of work and the multi-faceted shapes of cooperation that technology can put in motion (Schmidt and Bannon 1992). In this sense, the interactive requirement can be seen as a generative 'conversation' between the ideal and the enactment (Cohn et al. 2009); a middle, so to speak, between a constituted reality and one which is being made. The generative 'middleness' places requirements in a position of mediation that makes it difficult to attribute it with clear properties and transcendental values, but which considers requirements as moving and changing beings.

However, the interactive requirement can also be a hard 'thing', such as a contract, that can be talked about, articulated, and end up described. The interactive requirement, thus, is only related—not the same—to its descriptions, because these descriptions, however formal, are also performative (even if less so). Requirements, along with the conversation enacted by their descriptions, become a strategy that forms the world (not only information systems) in an intentional way, sometimes to the detriment of stakeholders. For example, Monteiro et al's (2012, p.577) study of the development of ERP systems in universities show how a supplier tries to "'smooth over' differences" in requirements between various implementation sites. Though requirements may be seen by some as being generative for generativity's sake, and thus, never-ending, the difficulty is that many requirements are complex because cooperative work is complex (Schmidt and Bannon 2013).

Though a never-ending exploration of requirements might not always be useful, either to vendors or even to stakeholders (e.g. functional creep), this elusive nature is telling. The question 'What is a requirement?' becomes groundless in light of its interactive potential. Contrary to the more formal engineering conceptions of requirements that consider them as tenable and stable, holding a certain value in time, the interactive requirement is inchoate, both in body and in purpose. Technology may demand functional closure (Kallinikos 2011)—for which purpose ambiguity is problematic since it can lead to the development of undesired interpretations (Stacey and Eckert 2003)—, but the agential relation of code, even if invisible, "is woven into the background of transactions, habits, and perceptions. . . " (Mackenzie 2006, p.170). Requirements, thus, just as code, and despite or because it hardens them further, partake to this weaving.

Some requirements, for example, will be designed with a 'handle' (e.g. a contract) to hold and define them, or to guide and impose a change. This often comes out from the more business-oriented literature that seeks an alignment between information systems and business goals (Tallon 2007). In other cases, the interactive requirement is too complex to 'handle': it becomes an entity fleeing closed understandings and engendering unclear consequences for the workforce and the relations between actors (Star and Strauss 1999). Though some requirements may be written down and legitimated by a contract, their reach can escape the rigid grasp. In this sense, an interactive requirement mediates changes beyond it conception, and even its definition is difficult to settle and changes over time (Darking and Whitley 2007; Lin and Cornford 2000).

If a requirement evolves in time, can be represented in several ways, and has changing relations to a generated reality (to which generation it partakes in), then only considering it conceptualisation as a 'hard' and whole object makes little sense. The notion of requirement, even if forcefully limited, is partial. The depiction of a requirement as a conversation and the mediating force it takes evades the single image of a requirement that would justly represent the world and the interaction of a machine. There is something alive that is not fixated and rooted in a unique definition that encompasses reality. Instead, a requirement becomes an abstract idea, a practical concept to which a multiplicity of descriptions sometimes hold down, other times hide realities, but which generally aim to create purposeful systems to re-organise at least some realities. This is, perhaps, the most notable evolution in the notion of requirements, which will be investigated further in this thesis: requirements do not necessarily represent the world (nor the machine), they generate new realities that are sometimes removed from the experience of its users (Orlikowski 1996).

To further this enquiry into the nature of requirements and depart fully from established concepts, open source and its peculiar treatment of requirements is a useful field.

2.4 Requirements in Open Source

Open source and *libre* software—software which code is openly accessible and modifiable—have grown much from the romantic ideals born in university labs (Raymond 2001; Stallman 1999). Open source now occupies a large part of the European Union's economy; some claim it contributes €450 billions per year (Hillenius 2012). New, hybrid forms of governance between community and commercial entities are becoming frequent (Bonaccorsi, Giannangeli, et al. 2006). The Linux kernel—the open source flagship project of the operating system—has both a large community following and important companies sponsoring employees to participate (Hartman-Kroah et al. 2009). Open source seems ubiquitous: beyond its software roots, some see it as a source of political critique of societal practices (Coleman 2009; Kelty 2004).

This section has one purpose: to emphasise the idiosyncrasies of open source in its treatment of requirements. Although the literature often confronts open source as the opposite of closed source software (Dinh-Trong and Bieman 2005; Noll and Liu 2010), it tends to forego explaining the reasons behind this opposition. It is in the context of these idiosyncrasies that the established notions on requirements can be unsettled to explore their nature. By escaping traditional concepts and putting them beyond their gravity field, the nature of requirements can be explored more fruitfully. To do this, requirements are considered under two aspects: sourcing and articulation. The sourcing of requirements deals with issues relating to their identification, while their articulation is concerned with issues of specification. Separating the two, even analytically, is no easy task: the literature shows both are interdependent (e.g. the form of organising affects which requirements are articulated). However, certain aspects of the sourcing and articulation of requirements need to be emphasised to present the particularities of open source (e.g. the cultural and political implications enacting a community).

2.4.1 Sourcing Requirements

From the labs of its political founders to the industry leading software, open source has made a disquieting trip. Open source was full of paradoxes (Lerner and Tirole 2002): why would anyone freely share code (Lerner and Tirole 2005)? Economically, reasons are usually categorised between intrinsic and extrinsic motivations depending on the participant type (paid or voluntary)(Bonaccorsi and Rossi 2006; Lakhani and Wolf 2005), with extrinsic motivations becoming more common (Ke and Zhang 2010). There is still ongoing research on the motivation of open source participants, and an impression that "participants are not economically motivated" remains in the air (Okoli and Oh 2007). In general, though, motivation research now concentrates on specifying the influencing terms and variables (Ke and Zhang 2010), or cultural implications (Subramanyam and Xia 2008), or the relations between motivational variables such as performance and leadership skill (Xu et al. 2009). However, focusing too much on motivation can lead to an oversimplified understanding of open source as there are many hybrid forms of open source software development, some involving closed source organisations (De Laat 2004).

Perhaps, a better way to describe open source is by considering it as a principally online community. Open source communities share with online communities the highly technologically mediated aspect of collaboration. This type of distributed collaboration is said to behave differently than in co-located spaces, especially concerning tensions (Hinds and Bailey 2003), though these may have positive outcomes (Faraj, Jarvenpaa, et al. 2011). This lack of co-presence has impelled some researchers to qualify this distributed organisation as virtual (Faraj and Johnson 2011), resolving the apparent contradiction of less presence (because of an increased virtuality) and more affordances for actors to enact control over routines. This is, perhaps, due to an increased visibility of knowledge and the actor's capacity to act upon (and profit from) its observation (Daniel, Agarwal, et al. 2013). This increased visibility and the way it is used has an influence on the formation of identities in new media (Vaast et al. 2013). A group's visible attributes would help crystallise its identity across various degrees of uniformity. In this sense, it would be possible to delineate 'them' compared to 'us'.

In open source, though, this kind of theoretical uniformity is limiting because participation in projects often escapes traditional lines of divisions: the ethnic backgrounds are diverse (Coleman 2012); those producing open source come from a large number of disciplines (Scacchi 2009); open source is another empire on which the Sun does not set harbouring different people (Colazo and Fang 2010); the meaning of open source itself is differentiated constantly through a number of performative repetitions (e.g. the explosion of Linux flavours that interpret what Linux should be or the contexts in which it is re-purposing itself) (Mackenzie 2006). Perhaps this diversity is due to the type of democratised innovation that propel open source (Hippel von and Krogh von 2006): usually heterogeneous; many times—at least at the start—innovation is inspired from individual and user needs (Hippel von, Franke, et al. 2009), and their diversity influences positively their success (Daniel, Agarwal, et al. 2013).

User needs are usually approximated to the interest that a project can gather around those needs. This has led to a number of studies looking at the institutional-economical aspects of open source communities, principally contributor attraction (Krogh von et al. 2012). The value created is dependent on the number of contributors that form that community. The dynamics of a community under this view are mostly dependent on the involvement of participants. Trust, for example, is an important factor in allowing this attraction to take place by enacting a "virtuous circle" of contributions (Benkler 2011). In these explanations, ideology takes a back seat; there is a logical reason why the dynamics of open source work: they allow the community to help each other on a common venture. The consequence of this (or maybe the cause), is often explained through similar arguments where a positive environment exists—allowed for by the materiality of open source—which enacts, in a structurationist fashion, institutional arrangements (Ostrom 1990). These norms are bolstered by the actions of actors who rationally decide upon their interaction with the project based on the outcomes of previous interactions (Schweik and English 2012).

Other authors provide a more sentimental, phenomenological account, Stallman himself, wondering about the machine's soul, compared proprietary software to an unpleasant limbo (Levy 2010). The community would exist, not because it makes individual sense to participate, but because it is the right thing to do. There would be, in this view, a transcendental value which would equate the machine's soul with a human's. These ideas are not so far fetched: Ciborra was already arguing for treating technology in a hospitable way as one would treat a stranger so as to "turn an ephemeral contact into a relationship which has the look (and the feel) of long acquaintance" (Ciborra 2002b, p.103). In this sense, there would be a certain symmetry (if not a complete one), between machine and man (Latour 2005). Open source here takes a prevalent position because its forms of organising are about *peer*-production (Bauwens 2009). Instead of the disconnected, far off machine, this peer-produced one feels closer to home, part of a larger assemblage of peer-actors.

This assemblage is often seen as a flat one, unconcerned with hierarchies (O'Mahony and Dahlander 2011). This might, at first, seem somewhat contradictory since open source is well known for its 'benevolent dictators for life' such as Guido van Rossum of the Python programming language (Raymond 2001). de Laat (2014) discusses directly open source forms of organising, comparing Mozilla's hierarchy-driven structure to new projects making use of (more robust) automated distributed tooling such as Git. Comparing typical open source software projects with Wikipedia, he concludes that the former still uses institutional hierarchies, only granting ex-post trust after participants prove themselves competent; while the latter grants trust ex-ante through peer-mechanisms. The difference between these two models is important to the discussion on socio-materiality. On one hand, the traditional open source model still provides an inherent value to the technology depending on the source of that technology; on the other hand, the Wikipedian model de Laat refers to does not systematically attribute such a value. Iivari (2010) comes to the same conclusion when she analyses power relations between user and developers to study the ways that requirements need to be asserted to the developers-as-gatekeepers (Scacchi 2009).

The previous paragraph might sound like a bleak view of open source, but if looked through the identity and online community literatures, open source communities provide an impressive development machine that is *actively* negotiating its ways of production. de Laat's (2012) findings of the bureaucratisation of open source shows the intense political arena that open source and other inspired communities have turned into. However, this bureaucratisation might come from the need to both cater for Stallman's four freedoms and the practical interdependencies of work in an open source project which prioritise which feature gets implemented first (Howison and Crowston 2014). Certain theories that account for these heated conversations are necessary to put the debates in perspective with the forms of organising. Such theories, like Faraj's et al. (2011) use of fluid concept, and Darking and Whitley's (2007) combined use of both fluid and fire concepts show to which point the foundations of open source are moving.

Open source has a wide diversity of actors involved which diversify the sources of requirements. The sources can escape the stereotype of the volunteer hacker as the only major participant (Asay 2012; Asay 2013; Capra et al. 2008; Lakhani and Wolf 2005).[6]. This makes general claims about the assemblages in open source dangerous. How can boundaries be set around open source? Should they? Assemblages is a good word to qualify the movements of collectives in open source

[6]Microsoft, for example, is one of the top contributors to the Linux kernel[7]

because it makes no judgement as to their nature. The symmetry between actors goes further than between man and machine, but enveloping different norms, cultures and organisations that open source cuts across. The range of applications and domains being entered by open source is growing (Fitzgerald 2006).

2.4.2 Articulating Requirements

The articulation of requirements in open source has been changing through the years. The foundational metaphor characterising development, the bazaar (Raymond 2001), has been moving forward to more nuanced "large semi-formal institutions with complex governance structures" (Coleman 2009). It is not just an informal place where each individual, as Raymond (2001) puts it, scratches his own itch. The development processes are a mix of political agenda and practical concerns. Far from dominating the individualistic and even libertarian pursuits are questioned by other conceptions of open source, such as the peer-production movement which has a larger communitarian dimension (Bauwens 2009). In this sense, trust and trustworthiness influence the way peers collaborate, form social structures, and articulate requirements (Benkler 2011; Crowston and Howison 2005).

In articulating requirements, open source projects have their own norms and hierarchies, usually done, but not always in an loosely organised way loosely organised team (Wu et al. 2007). Iivari (2010) shows how developers act as gate keepers for requirements. Users can become co-creators with the developers who endorse their requirements, depending on their own interests or the force of the user's arguments. The developers' abilities to push forward certain agendas also depend on their links to the project (Sambamurthy et al. 2012). The core ones have attained their central role through dedication and by gaining the trust of those who can give them commit rights to the source, representing a declaration of authority (Dinh-Trong and Bieman 2005; Jorgensen 2001). The ability to articulate requirements is thus not freely available to everyone and often depends on forms of hierarchy which can lack democratic transparency (Laurent and Cleland-Huang

2009).

The hierarchy is not established through contracts or salary, but by reputation and gift giving (Bergquist and Ljungberg 2001), thus, a developer that gives his time and implements requirements that are interesting to him or to the community gain reputation. Despite the existence of hierarchies, there is a latent emergence to the way requirements can be articulated. Part of this emergence is the sense of meritocracy that is seen to pervade development (De Laat 2014). Thus, enacting requirements depend on the willingness and the merit to implement them. Their specification takes a background role, helping to determine the possibility of their implementation (Howison and Crowston 2014).

Such meritocracy—to prove one's worth—depends on the availability of the source code. The openness allows for the *potential* of meritocracy to be distributed to unknown user-developers. This openness reaches far, influencing the articulation of requirements: it is because of the publicly visible contributions that core members claim and receive their legitimacy, which they then bestow on the requirements put forward by the users. The reason, perhaps, that open source is so generally viewed as informal is the large degree of self-assigned tasks (Mockus et al. 2002), although a number of core members have precise coordinating roles (Jorgensen 2001) and various open source projects recognise a number of formal governance models (Bacon 2012).

The emergent nature of requirements in open source is often attributed to various factors of stabilisation which allow for their informal articulation. In general, research understands requirements to be informal artefacts, pertaining to a (stable) 'vision' (German 2005), championed by established and core developers from user requests (AlMarzouq et al. 2005; Iivari 2009b), or deduced from the (stable) code (Rusovan et al. 2005). Requirements in open source form 'webs of discourse' throughout time and space, stabilising them into networks which gives them weight to be treated as connected wholes (Scacchi et al. 2006).

Putting too much stock on the perceived stable nature of requirements can be troublesome as noted by Di Gangi and Wasko (2009, p.1310): "Community members in their attempt to refine and clarify the idea for Dell created a situation where Dell misinterpreted user comments to determine which product offerings would be affected by the idea's adoption (the focus on business/work related systems only)." Freeman (2012) and Cornford et al. (2010) show well the temporal nature of discourses and controversies, and implicitly, how these influence requirements. The digital nature of the creative collaboration in open source attributes many meanings to requirements. Requirements which take a written form, such as in a mailing list (Scacchi et al. 2006), also carry emergent properties similar to digital, open cultural objects (Cheliotis 2009). The tools for articulating requirements are distributive and, paradoxically, centralised (Robbins and Feller 2005). Bug trackers, mailing lists, and version control systems help articulating requirements around a shared understanding of what features exist and which ones have problems (Weinstock and Hissam 2005). These tools help support an important quality requirement in open source: modularity, which in turn promotes governance models that fit distributed open development (Capra et al. 2008).

Open source takes a distinct route to engineering. Instead of siloing the different requirements tasks, responsibilities are often shared. Roles are separated usually by those who have commit rights, which is the main way to judge the legitimacy of participants in open source. The idiosyncrasies behind the forming of open source communities and the way these articulate requirements (both in an informal manner, yet with careful consideration to their place in the project (Howison and Crowston 2014)), show how alien open source requirements can be to their closed source counterparts, even if these can also less formal and methodical than usually described (Truex et al. 2000). Questions that would otherwise be tame in more stable and researched settings open up: what purpose requirements take in a project (Doganova and Eyquem-Renault 2009)? Why are certain requirements informal (Scacchi 2007)? How do they influence the project or the community (Iivari 2009a, 2010)? Do requirements evolve with in a distributed project (Cheliotis 2009), or do they stay the 500 page documents that they sometimes are taken to be (Heninger 1980)? In essence, questioning the importance of requirements in open source forces the enquiry of the nature of requirements and the weight they exert on the world.

2.5 Conclusion

Requirements have been conceived by the various paradigms in a number of different ways: first, as natural objects, clearly defined or definable. During the rise of structural analysis, they came to be seen in a more holistic manner as machine-oriented definitions of information transformation. Little beyond information transformation was considered, fomenting the impression that requirements could be controlled and manipulated. The machine and the world turned this idea upside-down by attributing to the world the materiality (as the possible material existence) of requirements, thus moving the orientation of requirements towards the problem domain. Soft systems went a step further: they turned the stakeholders into an analysts' peers and pushed a requirement's existence deep into a rabbit hole of questions; never ending and interpretively multiple, elusively escaping the cementing of requirements to only world-descriptions. Open source is portrayed as an alternative to closed source software, escaping traditional conception of requirements. Several elements, such as the heterogeneous and unexpected assemblage of actors, can help uproot those conceptions.

The evolution in thinking over the nature of requirements and the processes that cater to them has followed a path from complex but knowable requirements with clear techniques to represent them, to gradually consider them ambiguous and uncertain. Curiously, the evolution of requirements thinking resembles to some degree Mathiassen and Sorensen's (2008) work on equivocal and uncertain services. This is shown in figure 2.9 where the red line that represents this chapter's trip from section to section finally arrives at open source with an affirmation and a question. An affirmation because the literature places open source in a position to cater for specific needs (as itches) by allowing anyone to participate. A question because it is, as yet, how this participation takes place and the consequences it has on the nature of requirements.

The next chapter proposes an alternative theoretical understanding of requirements in a changing world, and de-emphasises the focus that is usually placed on the stability of the world.

```
OUTPUT_INTERFACE:  RADAR_ORDERS_BUFFER.
  ABBREVIATED BY:  ROB.
  CONNECTS TO:  RADAR.
  RECEIVES FROM:  DPS.
  PASSES:  RADAR_ORDER.
  ENTERED BY:  "MIKE RICHTER".

MESSAGE:  RADAR_ORDER.
  MADE BY:  RADAR_COMMAND.

DATA:  STARTUP.
  INCLUDES:  RO_ORDER_ID
             STARTUP_TIME.

DATA:  SHUTDOWN.
  INCLUDES:  RO_ORDER_ID.

DATA:  TRANSMIT_RECEIVE.
  INCLUDES:  RO_ORDER_ID
             RO_IMAGE_ID
             ALPHA_PHASE_TAPER
             BETA_PHASE_TAPER
             TRANSMIT_INFORMATION
             RECEIVE_INFORMATION
             NUMBER_OF_RANGE_GATES
             RANGE_GATE_INFORMATION.

ALPHA:  INITIATE_STATE_VAL_DATA.
  ARTIFICIALITY:  VALIDATION.
  INPUTS:  HANDOVER_DATA, HANDOVER_TIME.
  OUTPUTS:  UPDATE_STATE_VALIDATION_DATA.
  DESCRIPTION:  "THE REFERENCE ALGORITHM, KALMAN FILTER, IS
                INITIALIZED.  HANDOVER_DATA IS COPIED INTO
                UPDATE_STATE_VALIDATION_DATA WITH PLACE_IN_
                TRACK_TIME SET TO HANDOVER_TIME.".

ALPHA:  INITIATE_TRACK_ON_IMAGE.
  INPUTS:  HANDOVER_DATA.
  OUTPUTS:  HOIQ, STATE_DATA, IMAGE_ID.
  CREATES:  IMAGE.
  SETS:  IMAGE_IN_TRACK.
  DESCRIPTION:  "A REQUEST FOR PULSES IS MADE BY ENTERING A
                FORMAL RECORD REQUEST INTO THE HOIQ WHICH
                FEEDS THE PULSE SENDING PROCEDURES.".
```

Figure 2.1: Requirements in a formal language (Bell, Bixler, et al. 1977)

3.2.2.1 RADAR ORDERS BUFFER There shall be an output interface from the data processing system called RADAR ORDERS BUFFER. The data processing system shall communicate through this interface with RADAR. Across this interface shall be passed RADAR ORDER.

It is abbreviated by ROB. It was entered by MIKE RICHTER.

3.2.2.2 RADAR ORDER When transmitted across an interface the software shall handle the message RADAR ORDER. This message is made up of RADAR COMMAND.

3.2.2.3 STARTUP Information shall be maintained about STARTUP. This information shall include RO ORDER ID and STARTUP TIME.

3.2.2.4 SHUTDOWN Information shall be maintained about SHUTDOWN. This information shall include RO ORDER ID.

3.2.2.5 TRANSMIT RECEIVE Information shall be maintained about TRANSMIT RECEIVE. This information shall include:

RO ORDER ID
RO IMAGE ID
ALPHA PHASE TAPER
BETA PHASE TAPER
TRANSMIT INFORMATION
RECEIVE INFORMATION
NUMBER OF RANGE GATES
RANGE GATE INFORMATION.

4.1.3.1 INITIATE STATE VAL DATA Logical processing shall be done to INITIATE STATE VAL DATA. This shall have as input HANDOVER DATA and HANDOVER TIME. This shall have as output UPDATE STATE VALIDATION DATA.

NOTE: The reference algorithm, Kalman filter, is initialized. HANDOVER DATA is copied into UPDATE STATE VALIDATION DATA with PLACE IN TRACK TIME set to HANDOVER TIME.

In interpreting this requirement, note that the degree of artificiality in its statement is VALIDATION.

4.1.3.2 INITIATE TRACK ON IMAGE Logical processing shall be done to INITIATE TRACK ON IMAGE. This shall have as input HANDOVER DATA. This shall have as output HOIQ, STATE DATA, and IMAGE ID. This logical processing shall, when appropriate, identify a new instance of IMAGE. This logical processing, when appropriate, shall identify the type of entity instance as being IMAGE IN TRACK.

NOTE: A request for pulses is made by entering a formal record into the HOIQ which feeds the pulse-sending procedures.

Figure 2.2: Requirements in a natural language (Bell, Bixler, et al. 1977)

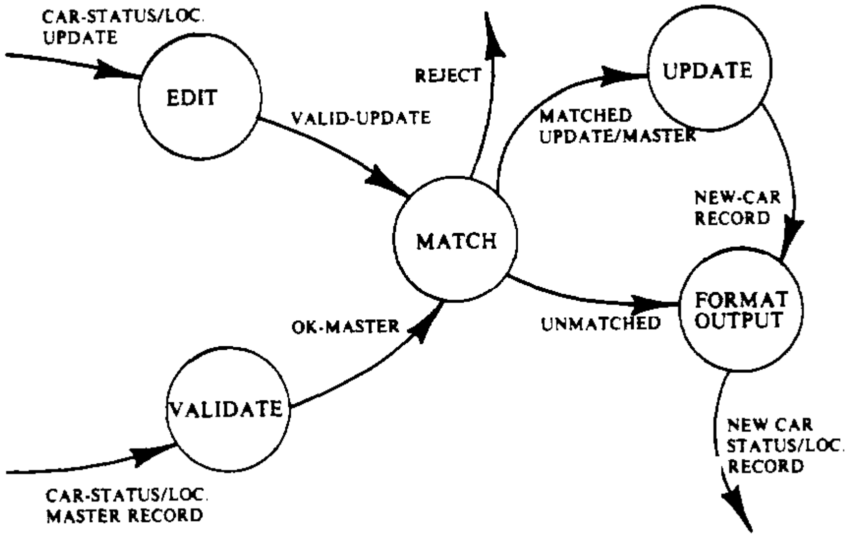

Figure 2.3: Requirements as data flows (Peters 1980)

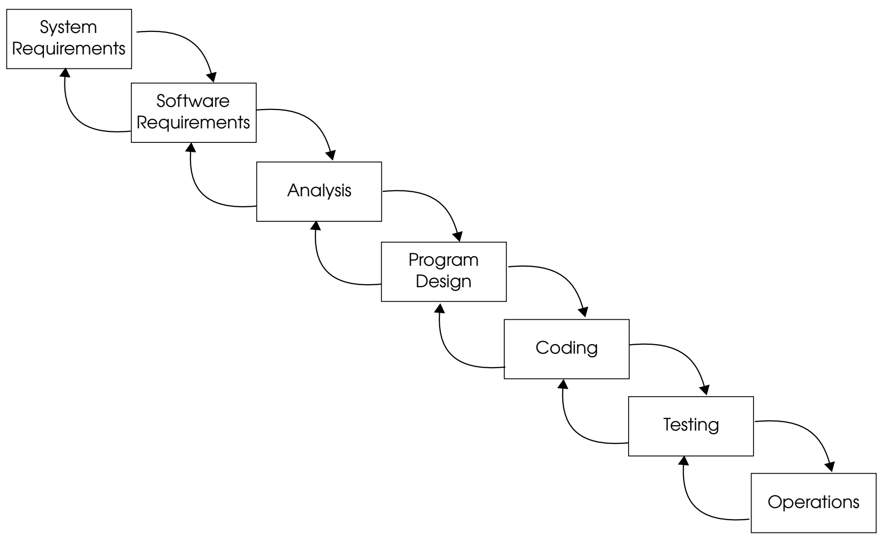

Figure 2.4: Waterfall model of software engineering (Royce 1970)

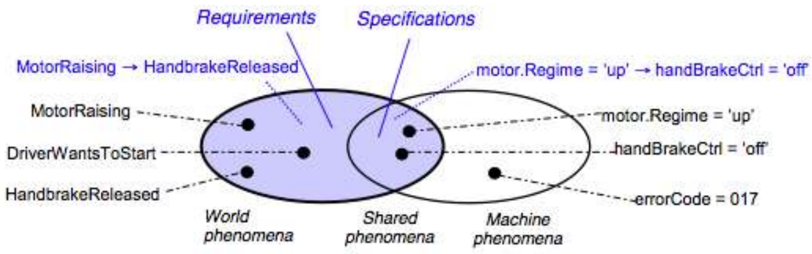

Figure 2.5: The world and the machine (Lamsweerde 2009a)

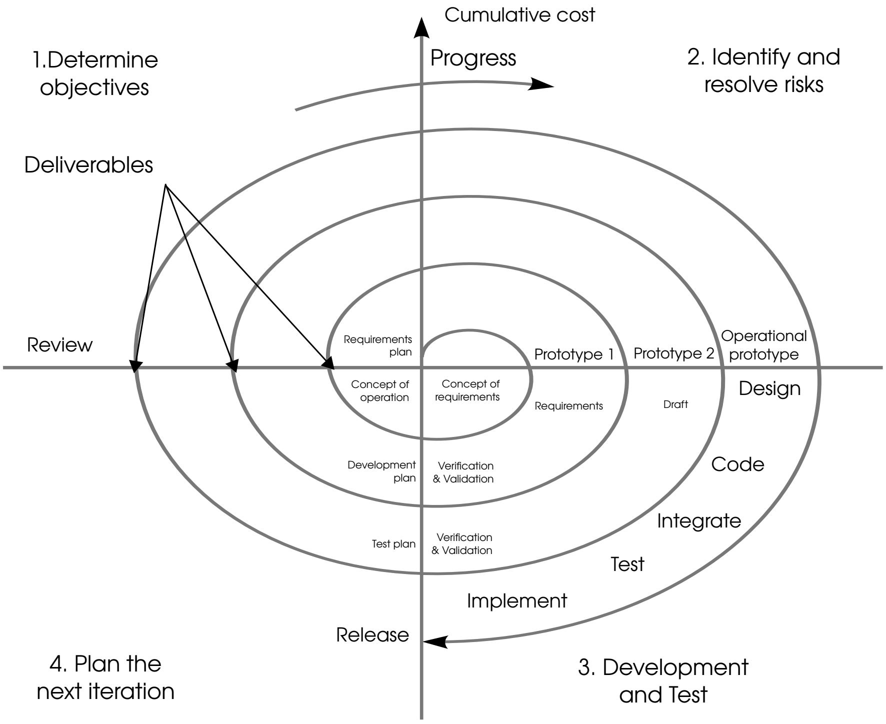

Figure 2.6: The iterative and incremental spiral process, adapted from Boehm (1986).

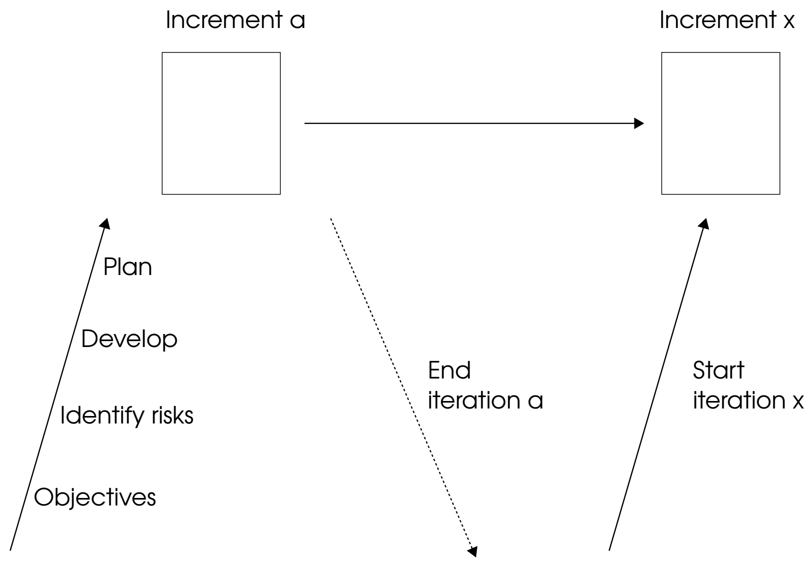

Figure 2.7: A transversal cut of the spiral model depicting it as an iterated version of the waterfall model.

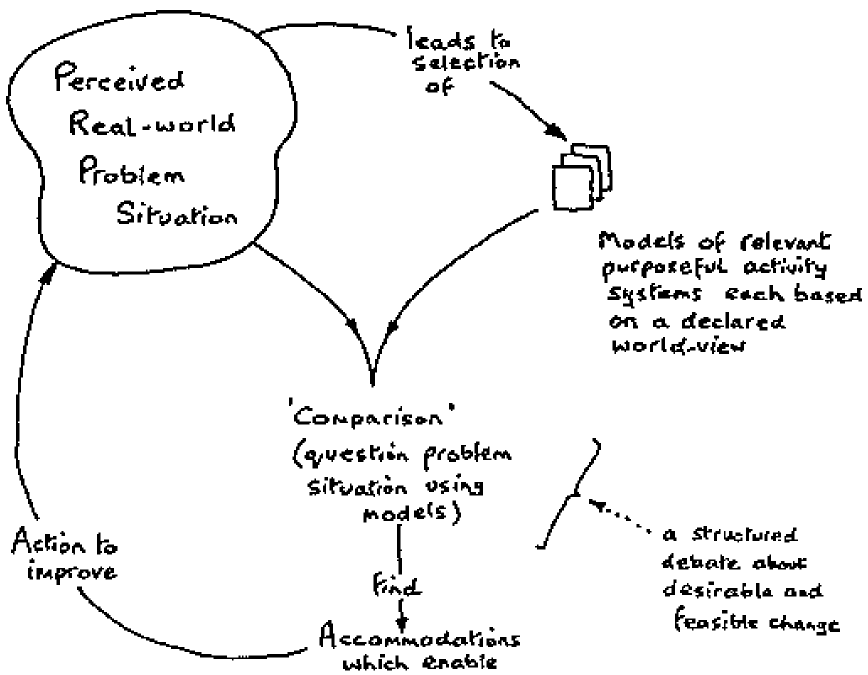

Figure 2.8: Activities in SSM (Checkland 2000)

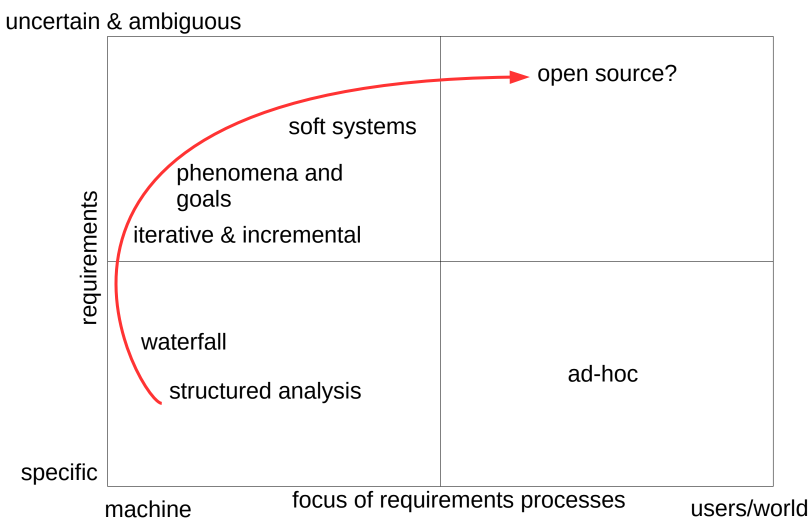

Figure 2.9: The chapter's red line showing the various ways in which requirements have been characterised by practice and literature. The line ends with a question: Is open source capable of catering to the ambiguity and uncertainty of the world?

Chapter 3

Theory

"Childhood is a geographical concept"—Jacques Brel

This chapter presents the theoretical framework used to investigate requirements in open source. The aim is to move away from a view of requirements dominated by notions of stability. The theory is founded on the questions identified in the introduction and given context in the literature review, especially those that discuss the role and purpose of requirements, and the assumed natural properties and functions which they are imbued with. Chief among these assumptions is that requirements are considered deterministic objects. Their role is taken to be straightforward: to simplify a complex world and to hold it down and take an accurate photo of it. That requirements can be both the picture and the camera at the same time challenges their assumed objectivity: they cannot be the method of representation and the representation itself because they would be the world. Once relieved of this mantra of neutral, objective power, requirements can be seen as political *actors*, used and using their influence to exert pressure on other actors and the assemblages they connect to.

Against these two notions of requirements as deterministic and objective representations of reality, and the corollary that they are stable or stabilising entities (since they are assumed to predictably represent the world), this chapter proposes the Deleuzian perspective that the world is in-becoming, and that change and stability have an intricate relation. In such a world, the statements and descriptions that are made about the world are also in-becoming, and thus, not necessarily deterministic.

The structure of the chapter is as follows:

The world is *in-becoming*. The ontological assumptions of a world which considers change as a normal state.

Requirements as rhizomes. Introduces an alternative perspective on the nature and attributes of requirements.

Unframing requirements. Ties the questions raised in the literature review and proposes a way to study requirements in a world *in-becoming*.

3.1 A World *in-becoming*

Much currency is given to stability in information systems. That the field thinks in terms of systems indicates an assumption that information can be categorised and bundled together, if not in nature, then perhaps in purpose. Information may be subjected, even tamed to fit certain analytic super-structures. There is no plural to information in information systems; the view is a holistic one aiming to explain phenomena along identified patternss. It is the prevalent view that there is a manifest stability that orders the world along describable structures. Whenever change exists or is acknowledged, it is to account for the stabilising efforts that actors may accomplish to (re-)enforce and legitimise orderly structures upon the world. Stability is the norm and changes are associated with temporal dysfunctions that actors need to remedy (Tsoukas and Chia 2002).

The consequence is that stability becomes the prevalent norm to understand how the world works. After all, the world makes more sense if it is explained through lasting institutions: winners write history. Stability creates the order that is prevalent in every institution. This argues that, without stability and the subsequent ordering, there can be no common vocabulary, no shared territory to base claims of requirements on. Intuitively, it seems, there is a pre-existence of objects, actors, networks, values, institutions, things that can be abstracted and grouped together; and these help make lasting claims about the world precisely because they are stable. Special status is given to those claims which in return receive a historical value. Biology became a science because it was able to categorise living institutions according to some general—and most importantly, accepted—rules. Stability works in aphorisms: a tree is a tree, a branch is a branch is a part of a tree. These are aphorisms because they are true by definition because they *are*, by essence, and this essence is opposed to other essences. For example, a tree is not a car because it is a tree since it has roots and branches. The essence, then, of a tree is to be alongside the other objects excluded from what makes a tree. The essence, therefore, is stable; it contemplates itself in a circle.

Because stability defines itself in a circle, it is difficult to reject, although, it is a fruitful tautology to question. What is it for something to be stable? The dictionary points to two definitions relevant to information systems. Stability either indicates a lack of change, or something that is firmly fixed. Faulkner and Runde (2013, p.60) say this when discussing materiality: "many if not most of the boundaries and categories we live by in our day-to-day lives are generally quite stable, at least relative to our life histories, and that the same is true of most of the objects classified within them (Kallinikos 2011). Otherwise, it would be hard to account for the apparent stability of social institutions and our artifactual worlds." This is an example of stability through lack of change, where social actors agree to a consensus on the role and purpose of certain objects, partly derived by the objects' materiality. Nonetheless, even lasting institutions, seemingly impervious to change or uncertainty, suffer from erosion. The lab fly—even if pertaining to a narrow and accepted category of biological genus, upon which numerous studies have been based and thus might seem stable—is judged by degrees of purity (Callon, Lascoumes, et al. 2011). Even Nature and its role are examined in debates which sometimes resemble schisms (Robbins and Moore 2013). Requirements engineering similarly wonders whether it has finally become a discipline instead of a craft

(Lamsweerde 2008). These debates reveal that even stable, almost institutional objects are discussed. They only appear to be stable.

On the other hand, actors sometimes take upon themselves to create change in an effort to retain or reach new forms of stability. Work-arounds, for example, have been examined as strategies to maintain a perceived loss of status (Zuboff 1989), or to negotiate relations between users and technology (Orlikowski 2000; Pollock 2005). Stability here is not a state, but a prolonged effort to stabilise relations. Work routines, for example, are continuously being re-created, supporting the idea that change is pervasive in organisations (Dionysiou and Tsoukas 2013). Change, as it aims to stabilise, happens in a specific context against which even stable, accepted objects such, as requirements can be, are confronted and disputed, making their existence different depending on the situation (Avgerou 2001). This view of stability undermines its prevalent position and puts change in charge of stabilising. For something to remain the same, there needs to be an effort in that sense. Consequently, stabilising and change become the object of study instead of stability, ultimately examining “the way in which reality is brought into being in every instant” (Langley et al. 2013). The result is an ontological readjustment with deep implications: how can a phenomenon be studied if it keeps changing?

Deleuze would answer this question by asking another one: how can a phenomenon not be studied in time? If objects are to exist, he would argue, they exist in relation to the world, seeking to create networks and assemblages. If something truly would not be changing, it could, at most, be a reproduction “according to which Being is repartitioned among beings following the rules of sedentary proportionality” (Deleuze 2004, p.355), that is, for example, by assuming “the same circumstances” (Deleuze 2004, p.4). But these ‘same circumstances’ would need to be assumed, and possibly reinterpreted in other contexts. Deleuze uses an example when talking about the annual feast of Bastille day which commemorates in France the storming of a prison by revolutionaries (Deleuze 2004). How ‘same’ is Bastille day year after year? How different? To this end, he argues that for something to remain the same, it needs to change in time according to rules that allow the reinterpretation of that 'sameness' so that the same can be recontextualised.

Change, according to Deleuze, is not only a question of time which enables change along specific rules (how fast or slow change takes), but also a question of relative space. Change alters the actors and their world. Changing work routines, for example, makes actors question how applicable they are in a new context. Applying this to requirements would be considering their relation to the world after a client changes their minds over priorities. Ostrom (1987) studied this aspect of change when she wondered what sustainable institutions in commons-based environments looked like. These, she argued, followed a variety of arrangements and rules that supported the involvement and acceptation of actors, re-enforcing the stability of the institutions in the commons. Unlike Deleuze, though, she gives agency for creating change to external actors only: "It is assumed that the momentum for change must come from outside the situation rather than from the self-reflection and creativity of those within a situation to restructure their own patterns of interaction" (Ostrom 2010). The problem with this view is that it imposes conditions on the capabilities of the actors. On one side, they bear an elevated status for their ability to make rational evaluations and decisions on change, but on the other hand, actors remain subservient to external change. Their only possible participation is limited to individually deciding whether to form part of the commons, or to pull out.

Deleuze and Guattari oppose assemblages to strata (though they are produced in them (Deleuze and Guattari 1987, p.557)). Strata are formalised and established groups that over-power the creation of meaning, such as an institution would. The opposition stems from the different logics found in both: whereas in strata, things are well-ordered and the meaning is sedimented, in assemblages things and their meanings try to flee along lines of flight. An assemblage—*agencement* in French—is a grouping that does not have essential (independent, pre-existing, or eternal) values (DeLanda 2006). In this sense, there are few limits to the composition of the assemblage, at least while it is constituting itself. The way the assemblage forms (e.g. by trying to settle debates), influences what the grouping can do, and what becomes included in the assemblage, which is why *agencement* is close to the word agency.[1]

On the question of the agency of change, Deleuze and Guattari sometimes approach phenomenologists. Change is not an external factor, an alien invasion that alters the world. Change is, paradoxically, much more permanent. It is almost a condition of the world. There is a certain drift to it that makes the world difficult to grasp: there is no hierarchy, no evident rational for change to take place. The world under this view does not come with a handle. Instead, the world needs to be apprehended by coming into contact with it. Actors often act on the world in ways that escape normalised behaviours: they improvise. They take responsibility for the changes they make. Depending on situations and how actors choose to make sense of the environment, they will develop their own solutions which are instantiated to the context at hand. Weick argues: "People make sense, try to introduce order, and then selectively single out manageable moments from a vast undifferentiated background. When people 'introduce' 'order', there is no guarantee that it will persist"(Weick 2006). The world seems largely indescribable, until actors arrange it in some way.

Actors, then, are not just recipients of change, but actively participate in enacting a world by making sense of it. They consider the world under their own ethical system. Making sense of technology, is akin to "welcoming a stranger", critically evaluating the technology that came into the human's house (Ciborra 2002b). Or did the human enter the technology's house? This is where Deleuze and Guattari distance themselves from phenomenology to enter a much more networked world where categories and boundaries are less evident. A human is not only a human anymore in so far as he is caught up with processes: the human, or some of its qualities, is part of a transformative phenomenon. The entanglement becomes so rich that it is difficult to distinguish what part of the process is crucial to it. They

[1] In simple English, assemblages are groupings of entities forming vulnerable systems in which meaning is (actively) open to change. They are vulnerable because they lack the dominating power of established systems which close around their well-defined signifier. The reason why assemblages of requirements is an interesting theoretical construct is that it escapes the established meaning of requirements, allowing them to form relations with entities that are usually ignored (such as identity or territory). Deleuze and Guattari put the identity of entities in a paradoxical position of freedom (identity is in-becoming) and crisis (it is not stable, its existence is volatile). Open source can project this crisis by altering the established knowledge on how and what requirements should be.

argue: “There is no such thing as either man or nature now, only a process that produces the one within the other and couples the machines together”(Deleuze and Guattari 2004, p.2). For Deleuze and Guattari, the way to explain a changing world is to escape the use of confining categories such as object-subject. Defined categories can only help reproduce the same institutionalised explanations. They hold instead that transformative change can only be realised through changing intensities: “everything is objective or subjective, as one wishes. That is not a distinction: the distinction to be made passes into the economic infrastructure itself *and into its investments*”(Deleuze and Guattari 2004, p.378). It is through these ‘investments’ that are of an existential nature to the actors, that the world in change can be understood. By casting aside already defined categories and perspectives, it puts the actors as enacting a world, while being the world themselves, without any distinctive properties. In other words, actors invest *themselves* in the change that is taking place. In this sense, the world is not changing at the mercy of some external pressure, but is *in-becoming*, part of an unknowable number of transformative fluxes that are of deep concerns to the actors.

What concerns motivate actors to enact their world *in-becoming*? Why would any actor want to necessarily stabilise it? Why not destabilise it to make something else? Actors behave in ways that satisfy their desires. These are not necessarily conscious, but they are intrinsic to any actor. Since the world is *in-becoming*, and actors compose the world, they are themselves *in-becoming*. Deleuze and Guattari, just as Latour, place no distinction on what constitutes an actor. As Massumi puts it: “‘real’ hunger is as much an economic reality as a digestive fact” (Massumi 1992, p.94). In fact, an actor could be considered an object for all intents and purposes; and actors or objects themselves cannot be distinguished from the world, so deeply involved in the world are their desires. Instead, what can be mapped are intensities with which actor-worlds evolve. This blurs the contours of traditional categories and allows actors, whatever they are, to form heterogeneous assemblages that are linked through their desires and the transformative capabilities they provide. In this sense, Deleuze and Guattari believe in existentialist worlds, where assemblages of actors follow through their own, vital necessities and explore their desires.

The blurring of categories that Deleuze and Guattari advocate when talking about the world induces a certain schizophrenia about what anything *is* since everything is *in-becoming*. From their willingness to escape already-made definitions, Deleuze and Guattari, instead, propose to reconsider the transformations that take place. Actors, and their assemblages, desire to become other things, to evolve, for whatever reason. This engagement from the assemblages forces the study away from stable entities conforming to their state, but embraces the movements and countermovements taking place *in-becoming*. The reasons behind the changes become more personable, anchored to motives that are close to the assemblages and actors. What does it mean for a requirement to be open source? Why are open source requirements processes associated with democratic values? Why is there a need for a process to be standardised and evolved through reality? Considering the movement of things allows for the type of questioning that does not assume that something is just one thing or another, but a complex mash which evolves through time. What interest Deleuze and Guattari are the circumstances which produce movements (Massumi 1992).

This complexity, this tendency for things to be always *in-becoming* is a challenge to the typical understanding of requirements engineering, where requirements are grounded, formalised, objectified. It is also a challenge to the way requirements are spoken of and spoken for. It is only a little stretch to say that a requirement is not real. Should a requirement that is written, passed around, communicated through the team not be considered one? Would this not be a negation of actors and assemblages to hold certain properties, to carry certain weight in specific situations? For example, could not a requirement be seen as a contract by contractors and as a boundary object by the programmers (Barrett and Oborn 2010)? In other words, does a requirement not have substance? That something is *in-becoming* does not mean that it is volatile, becoming one thing or another randomly. On the contrary, assemblages respond to their own desires according to their own logic and language. Things are not born out of a vacuum; they possess an intrinsic vitality that they manifest through connecting assemblages. This potentiality is virtual and can only be perceived in movements of varying intensities, when causalities become evident. Before anything is *in-becoming*, a certain potentiality exists that, if let to its own designs, can evolve, combine and produce other assemblages. Meaning for Deleuze and Guattari, according to Massumi, is a contraction of time, combining traces of the past, interpreted in the present and allowing for potential futures (Massumi 1992).

Assemblages are themselves productions, formed through transformations at the hand of a variety of actors, also produced and transformed by assemblages. The actors, and the assemblages they engage with are as varied as "machine productions", "aesthetic practices" and "time" which can be "activated, orientated" (Guattari 1995). These movements, and their study, can bring about "new constellations of Universes of reference" by re-centering the understanding of the world to the study of the circumstances (Massumi 1992). The subject, traditionally envisioned in opposition to the object, does not exist anymore. It is part of a transformative flux of productions that *subjectivises* (Guattari 1995). In this sense, the subject is not entity x, which is not equal to person y, but a series of circumstances that have combined to enact a certain action. There is therefore, no necessary or obvious logic to why something is something in a world in-becoming, because the way statements are made is not through the identification of what an entity is or is not, but the movements it has in the world and the connections it enacts. In a world in-becoming, a requirement is not defined as only a piece of contract, or only a statement of intent, or only a piece of paper holding a user story; a requirement can be, above all, something which has sufficient force to alter the movements of the project or other assemblages with which it engages. To study requirements in a world in-becoming is to study how requirements, as a force, travel and create change for whatever purpose its various assemblages have intended. This force can be quelled and used to pacify energies, or it can be used as a cartographer would a map, to find and discover new, unimagined territories.

3.2 Requirements as Rhizomes

Although stability is still the ontological paradigm prevalent in requirements engineering, perhaps due to its still recent advent as a discipline, some ground has been covered towards a more balanced view of change. The notable rise of goal-oriented requirements engineering, the purposeful exploration of multiple choices of systems, the autonomous agents that influence requirements. Such research opens the door to progress beyond an ontological dichotomy of a world itself divided in its subset and a machine. To move further beyond this dichotomy is to admit that there exists multiple worlds and multiple machines with a relative influence that still needs to be studied.

To talk of multiplicities is to cast aside the prevalence of stability in the creation of information systems. It is a first step in imagining that stability is the outcome of change; that it is stability that is the temporary dysfunction, not change. Stability taken as static is a singular glimpse in a sea in movement. The Popperian view that scientific paradigms are stable only after a moment of instability in which previously prevalent understanding are falsified puts the falsification process as the initiator of stability. Change, therefore, has a significant agency. "The multiple *must be made*," exhort Deleuze and Guattari (1987, p.7). The book, much like a requirement, can be seen as an object, suspended in time and holding a discrete amount of information. This is what Deleuze and Guattari call the 'root-book' (Deleuze and Guattari 1987). The 'root-book', they argue, is simplistic because it is taken away from the world that created it. In essence, it is 'othered', made a stranger, objectified away from the assemblages that have created it. By objectifying itself, the 'root-book' removes its moorings from the world to contemplate its own subjectivity. The subject has been objectified, the requirement has lost its legitimacy only to acquire an artificial one: it has become a contract, "an image of the world" (Deleuze and Guattari 1987).

But the book and the requirement can be something else entirely. They can be a 'multiplicity', *made* to assemble themselves into worlds according to their own logics. Such a book would not be discrete, it would have a movement depending on its moorings with the world. Viewed this way the book as an object would be endowed with an agency greater and lesser than its own: it would not look to itself but would be considered alongside its moorings. Just as the book is not created in a vacuum, so it is with requirements. The world in which they live, according to Deleuze (2005a, p.38), is a "world of exteriority, a world in which thought itself exists in a fundamental relationship with the Outside, a world in which terms [*i.e. objects*] are veritable atoms and relations veritable external passages; a world in which the conjunction 'and' dethrones the interiority of the verb 'is'." Latour would call this impossibility to reduce actors to simple self-contained entities the irreducibility of the actors.[2]

These relations—called moorings earlier—are material to the worlds. They are assemblages of objects which combine together to produce something. The objects do not control these assemblages, they are "external and heterogeneous to their terms [*i.e. objects*]" (Deleuze 2005a, p.37). There is, it seems, a paradox between the agency brought back to the objects, and their definition through the contexts to which they are assembled. The contexts, to a large extent, define the object-moorings assemblages. The book is put into perspective: it is read, translated, interpreted, contextualised, applied, refuted; it belongs to a time-period and beyond, movements of thought, spaces of action. The requirement, entangled as it is to its process, reflects the world, is debated, changes forms and becomes something else entirely. It is hard to grasp without a specific intervention to settle it down, achieved only through its rooting to an objectified form. But it is this difficulty in its grasping that allows the requirement to be something else entirely than it could have been, to escape its categories and become something, perhaps unexpected, but very much relevant to a certain context.

There are structures in nature not unlike the object-assemblages described earlier. The rooting resembles the tree, the assemblage resembles the rhizome. The tree spreads by copying itself. The branches, the roots, the leaves; all belong to the same system. The entire tree can be traversed in polynomial time, and all the

[2] In this section, I use the term 'object' interchangeably with the concept of 'actor'.

trees can be traversed in this same way. The rhizome, unlike the tree, is a bulb that does not follow an order in its growth. It spreads (seemingly) randomly, through no discernible pattern. It cannot be traversed by simply visiting all the leaves and reporting back to the branches to then report back to the tree. The rhizome has no such reproducible hierarchy; there is no unitary, sequential hierarchy. It is *in-becoming*: if a rhizome is cut, it will grow again, unpredictably, along the spaces it has visited. Rhizomes have no specific shape and therefore escape causal explanations. For Deleuze and Guattari, their development is analogous to the many ways that things evolve: by changing the arrangements of object-assemblages (Malabou 2012). Requirements as rhizomes would not look at their objects, but to the ways in which they have evolved and could evolve *in the contexts of their assemblages*. To understand a requirement, therefore, the requirement must be taken into consideration along the forces that shape its object-assemblages. The requirement gains back its agency at the same moment that it loses it. Entangled in its concerns, the requirement alone cannot answer these questions: why is such a requirement? when is such a requirement? how is such a requirement? A requirement *is* not, but it is more alive than any that *is* because it is evolving.

The evolution of object-assemblages for Deleuze is curious; it does not necessarily follow typical constitutive properties. Rather, an important aspect of the evolution of object-assemblages is their ability to become beyond that which is initially thought possible. Object-assemblages are not independent of the world in which they evolve. Object-assemblages, Deleuze argues, can also distribute themselves in an "open space" according to nomadic logics where "nothing pertains or belongs to any person, but all persons are arrayed here and there in such a manner as to cover the largest possible space" (Deleuze 2004, p.46). There is a viral quality to this type of distribution where boundaries are, if set, temporary or indeterminate. If boundaries exist at all, they are empirical and can change in accordance to non-linear, seemingly random logics. In other words, the world cannot be carved out by hypothetical boundaries traced by an actor wishing to impose its own logic. The order of the world and its construction is much less arbitrary and depend on the movement of the multitude of object-assemblages and the territories they visit. These false boundaries have much less agency than in more static ontologies—if they have any—in this open space; they do not determine how the world is defined. The world is, as it were, up for grabs by the object-assemblages that wish to distribute themselves inside. This is important for an alternative understanding of requirements: that requirements are not constituted objects, but that they can be constitutive of the project and the project constitutive of requirements. Where the requirements as a rhizome stop and the project begins is not so easy to determine.

It might seem that Deleuze does not believe in the existence of properties, that requirements are only empty shells that fill over time according to the will of their assemblages (Smith 2012). After all, his understanding of the world is that it is always premature, always in need of creation. Even though Deleuze might seem to approach certain phenomenological tendencies—his admiration for Spinoza is telling—his adoption of phenomenological concepts concerns the way assemblages form. The properties of actors, such as requirements, can only be understood in their full sense when in a assemblage. Deleuze says: until the flowers are there, the Sun is not the cause of the flowers. It's only once the flowers have appeared that the causality of the Sun becomes evident (Latour et al. 2011). This example is strangely paradoxical and almost pushes Deleuze into positivism, were it not that it is an anti-causation. In other words, the condition might seem causal now, but for it to have become so, much effort has had to be made. There is no direct cause between the Sun and the flowers, there cannot be such a link only until these two actors have entered each other's assemblages through the mediation of intermediary assemblages. For Deleuze, this assemblage is not necessarily new, but is in-becoming. Just as much as the bee becomes the object-assemblage bee-flower (that is, the bee is attracted to a flower in particular, at this moment, despite—or because—of the coexistence of other flowers), the flower has desired to become as such so that it enters the Sun's object-assemblage. But it is not only the flower's desire, but also that of chlorophyll, and the tissues that support it. This marks a principal difference between Latour and Deleuze. In the flower example, Latour would claim that chlorophyll has been invented the moment when the Sun and the flower enter into their object-assemblages, whereas Deleuze would have argued for the existence of that potentiality for such an object-assemblage to have already existed beforehand. This is not to say that things exist as such since they are in-becoming. The flower and the Sun are, to all extents and purposes, in-becoming; the flower opens its head to welcome the Sun in the morning and dips it when it departs; it fights with the other object-assemblages for the Sun; and so on. To apply modern information systems language to this example, the flower and the Sun have a materiality that gives rise to a particular materialisation that is in process (Leonardi et al. 2013). Requirements as rhizomes hold traces of the past as substance which can take form in the present and which helps define its potential. Materialisation, therefore, is partly dependent upon past events.

What Deleuze, and later with his colleague Guattari, argue, is that materialisation is *in-becoming*. It is not materialised, but in-, and out-materialising. Some properties of the assemblage might be in the process of stabilising, but there are efforts, desires in place to materialise these properties in time, and also to evolve them to the new contexts not present when the original experiment took place. A flower will always need to be *in-becoming* with its Sun: it will need to repeat that first original experience that made it evolve to what it is to be *in-becoming*. It will cast its head up, looking to the west, it will send its roots downwards and protect itself from being eaten. The gardener will look to it, and so will the bee that uses it and is used to further repeat that original experiment in time. When the original experiment is repeated, it is translated to the particular context in which the flower finds itself. In that sense, the experiment is not transposed, like a scientific experiment is often thought to be, unless it makes the purposeful assumption that the same conditions perdure. For the same conditions to perdure, a taxing process must take place. It is, perhaps, at this point that Bourdieu and Deleuze converge only to diverge. If both acknowledge the existence of structures to reproduce a certain order, Deleuze will argue that there is no pure reproduction (Deleuze 2004; Deleuze and Guattari 1987). In other words, that *the* original can never be made again because it will either be too specific and thus not reproducible, or too general and thus, as well, not reproducible in the specific. Bourdieu escapes this problem by arguing the reproduction of certain stable structures which are repeatable, translatable to various contexts through rituals that aim to preserve an identifiable core identity (Bourdieu 1992). Deleuze does not refute this but argues that there cannot be any repetition without differentiation (Deleuze 2004). Given the intimacy between the repetition and difference, Deleuze concludes that "if repetition can be found, even in nature, it is in the name of a power which affirms itself against the law, which works underneath the law, perhaps superior to laws. [. . .] In every respect, repetition is a transgression" (Deleuze 2004, p.3). The rhizomatic requirement, is thus, never an exact image. It is translated, not only to adapt to new environments, but also to maintain that 'sameness', whether of itself, or the assemblages it engages with.

To repeat, to conserve that 'sameness' is not as easy as it sounds. Weick's example of jazz is helpful here to explain the rebellious desire that repetition has, and by contrast, the appetite for prevalence that difference is keen on (Weick 1998). Deleuze argues: "[t]he fact is that specific difference is maximal and perfect, but only on condition of the identity of an undetermined concept (genus). It is insignificant, by contrast, in comparison with the difference between genera as ultimate determinable concepts (categories)" (Deleuze 2004, p.41). Improvisation exists because there is a genus identity to the concept of jazz. Jazz is unlike classical music because their genus are unlike, but rocky-jazz are alike both rock and jazz. It is the indeterminacy of the jazz genus which allows improvisation through specific difference of repeatable concepts which are recontextualised. A concept which is both repeatable and differentiable because of its allowance of becoming. How else to interpret the phenomenological, and sometimes mystical, explanations given to practice (Zuboff 1989)? Repetition, in many cases, is not a matter of doing the exact same things under the same conditions, but to improvise through differentiation.[3]

[3]Or as Deleuze would say *differen-c-iation* in French, that is, the mathematical process of deriving multiplicities. He argues for the mathematical deriving which, contrary to the typical definition, can be defined as its own element.

Improvising is not a simple act. It may, at first, seem like an action subordinated to an earlier one requiring little or no agency from the improvisers; an automatic response to a stimulation. In many cases, "competent improvisation involves a registering of, and intervention in, the situation that seems for the outside observer almost out of context, since it is fast and unexpected"(Ciborra 2002b, p.49). There is an ungraspable quality to improvisation which is not unlike Deleuzian and Guattarian concepts. Similar attributes of unexpected contexts (new situations), multiplicities of meaning and potentiality (sense making), and becoming (time) sound similar. These notions indicate an abrupt rearrangement of an established order, and in so doing, an alternate reality which responds to its owns speeds and boundaries. What is normal in improvisation 'mode' is abnormal when acting out routines. A requirement in rhizome speaks its own language, trying to satisfy its existential desires by combining with other assemblages.

Although a rhizome holds some traces of the past, it is in the present that these properties become evident again. It might seem that Deleuze and Guattari believe that the world is not made because it never reaches a sufficient maturity to stabilise. Deleuze and Guattari do not reject stability, they acknowledge its existence but confront its supremacy. Repetition, often erroneously equated with resemblance is shown by Deleuze to be an essential part of differing since "repetition is in essence symbolic; symbols or simulacra are the letter of repetition itself. Difference is included in repetition by way of disguise and by the order of the symbol"(Deleuze 2004, p.19). In other words, if patterns exist, they can only be reapplied through translations with regards to the original. In this sense, patterns can only be enacted and never abstracted, since their abstraction (if they were ever to reach that level of purity) would make patterns unrepeatable (because undifferentiated), attaining the divine coating of moral law. Therefore, when a requirement is applied in a context—it necessarily and always comes from another context—the requirement is questioned: 'what remarkable aspect makes the requirement able to travel through contexts?', 'what remarkable context apprehends this other one?'. This is another reason why requirements are always in-becoming: they are always Other, multiplicities of themselves, made different through repetitions.

The requirement as a rhizome is Other because it is an irreducible object: its enacted self can only be apprehended from its becoming with the world and the assemblages they shift. This is, perhaps, the reason for the multiplication of requirements methods, all in their number trying to apprehend that ever-elusive Other-requirement that travels nomadically before their eyes. Requirements are the uncanny Other, little resemblance and disguised difference (Freud 2003). To understand why a requirement is, repeatable, yet contextual, it is necessary not to abstract it, but to *differentiate* it, that is, to give it attributes that allow it to derivate.[4] The Other, therefore, is Other in its own multiplicity. In other words, the Other, because *and* despite the Same has its own intensities, its own assemblages. Zizek says this of Pollock, an artist he claims is "the ultimate Deleuzian painter" (Zizek 2012), perhaps because "Pollock's contribution to the evolution of art is secure. He described Nature directly. Rather than mimicking Nature, he adopted its language—fractals—to build his own patterns (Taylor 2002)". Pollock's art is different to Nature *because* he does not mimic it, *despite* adopting its language. Here, Deleuze confuses the issue some more by theorising both difference (the because) and repetition (the despite) on their own terms *and* together.

Applying this to requirements would be saying that a requirement can only be distinguishable through repetition. That is, what makes a requirement, the values that it encompasses, the projections that they give rise to, is not the work of a contract, or single identifiable objects, but that it is *made* in time. For a requirement to become a project's requirement, it needs to be induced with some specific properties that are general to the project. This induction is far from an evident application of values and qualities, but a making sense of the project and its place in the world. In practical terms, this means that a requirement would owe its being to the arguments, the worlds, the creations that are making them. That is why requirements would be in-becoming, because they are both repeated and

[4] I use the term 'derivate' instead of 'drifting' that Ciborra saw as more translatable into English because it does not have the connotation of passive agency. It might seem that a 'derivate' is a state, but since by definition it is an approximation—i.e. it is not just a thing but a thing and its movement—it does not contradict its natural becoming.

distinguished, instantiated and universalised. Requirements need to ‘fit’ in a project, not through reduction, but through the intensities they emanate, the new worlds they connect to, the alliances they allow.

This does not mean that every requirement is *in-becoming*. Not all requirements are so alive. Usually, the better they are described, the less becoming they have since their description is the result of a stabilised consensus. The described requirements are more plain and evident. Their role is different when questioning the project any further because they lay the ground for what is acceptable later. Essentially, they allow certain kinds of debates and close others. They do not manipulate the world, and make no direct change to the project. In a way, they are taken for granted. This type of requirements is the book-tree discussed above, they are describable, known and finite. There is little entropy, controversy, or becoming, yet, they serve a role in conjunction with requirements *in-becoming*. A requirement that is more than a statement is necessarily a rhizomatic requirement *in-becoming*. It has potential, in a timeless manner in which past, present and future routes converge making it a multiplicity, enacting itself in ways that might escape any logic except its own. Thus, it evolves in combination with its assemblages.

3.3 Unframing Requirements

Studying requirements in a world in-becoming demands the rejection of certain assumptions often taken for granted and discussed in the literature review:

- Requirements are obvious
- Requirements pre-exist
- Requirements are describable
- Requirements are bounded

Examining these assumptions from the theoretical perspective of Deleuze and Guattari rather than from the literature provides some indications as to how else can requirements be studied. In turn, rejecting these assumptions necessarily lead to accepting other assumptions.

Some requirements may seem obvious at first, but may be more complicated. An obvious requirement may be simple because its enforcement is simple, being pushed forward by an effective interest. This may be the case, for example, when some groups of clients hold more power than others and are able to press their affairs. In other cases, a requirement may be simple because it is evident, but the implementation of which is a matter of discussion. In health systems, the Hippocratic oath 'do no harm', for example, is a simple requirement, but one that proves difficult to implement when questions of privacy and security come into play. From a Deleuzian perspective, obvious requirements can exist. However, they can be an indication of a lack of becoming in the project.

Requirements are often assumed to pre-exist. Requirements are *elicited, found, captured*, even *trawled* (Cohn 2004). Consequently, requirements are objectified by already having a pre-existing, preternatural life. They exist to be found, yet are in such a form that explicit techniques need to be put in place to capture them. Such requirements, even though they exist, can only be obtained through indirect methods of observation, giving them an objective aura. Objectifying requirements runs the risk of imposing certain properties and weights to certain requirements artificially, constraining what could otherwise result in a fruitful exploration of the world. A pre-existing requirement assumes that it has no space to grow, no controversial aspect that might lead to other solutions through exploration.

To fully describe a requirement would mean that it has no space or time to move. If combined with the other assumptions, this means that a requirement, as soon as it is discovered, is found in its final state and is incapable of evolving. On its own, though, this assumption adds to the preternatural quality that requirements are imbued with, essentially objectifying the requirement. Any further description attempt is only meant at a reification of the requirement's natural description. This assumption has been criticised by agile methodologists, who, instead, prefer to see requirements as part of an evolving understanding of the project's complexity

(Beck 1999). A describable requirement, coupled with the belief that they pre-exist, might lead the project to reject other desirable routes of exploration.

Requirements engineering considers important placing boundaries around the world. This limits the requirements to be found by delimiting the vastness of the world. Creating boundaries defines a legal domain of exploration as opposed to illegal, illusionary requirements. The result is that very early on, the project is well defined *only against* a few design ideas which are seen as coherent at the onset of the project, and not necessarily aligned with the world. From then on, the exploration tends to be top-down, grinding those design ideas into requirements. Any bottom-up action is a chimera, since the bottom-up is limited to the frame already defined. The bottom-up exploration is thus conducted to fit against a list of requirements choices which hardens through continued, limited explorations. To the Deleuzian viewpoint, this is a loss of wild energy that could otherwise be used to combine the project with new worlds. Instead of exploring, the project narrows itself into definitions of what it is, or what it is not, through simple logics of identity: project x = project x != project y. A Deleuzian approach would open up the project's most basic aspects to the interpretation and recombination to other possible worlds of use.

To show requirements are *in-becoming* amounts to show that there is no definite state or stage for requirements, or their processes. Instead, requirements become part of a larger process of transformation, which aligns to this or that assemblage of goals. From the moment requirements stop being seen as only system descriptions, they stop being cold artefacts, objectively ordering the world. Once their reach starts being evaluated their participation in complexing the world can be gleaned. The result is a disparity of types of requirements, definable by the worlds they touch and those they destroy, just as any creative enterprise is always also destructive (Malabou 2012). If a requirement describes one thing, it stops describing another; an opportunity cost is always involved as opening a world means closing another one. Exploring which requirements are needed therefore has consequences on the identity of the system that is to be created. Massumi provides the analogy of the spatiality of rhizomes fitting well with the study requirements: “which is reached when circumstances combine to bring an activity to a pitch of intensity that is not automatically dissipated in a climax leading to a state of rest” (Massumi 1992, p.7). Under this light, a requirement would be a series of circumstances which create various possible routes that could alter the identities of the actors involved.

Rejecting the basic assumptions of bounded, pre-existing, describable world necessarily involves agreeing to another: the Deleuzian perspective assumes that there is a way for requirements and their processes to be in-becoming. This implies that, either the capabilities of the actors involved allow for this amount of change, or that the materiality of the program or project under construction is flexible enough to allow for changes in its identity and purpose. Ultimately, this means that requirements, and what is understood by them, need to be amply pliable both in their materiality and their materialisation to allow for alternative purposes, roles and identities. In a sense, what needs to be shown is that requirements are more than a statement.

Requirements are taken to be so many things, that it is not too extravagant to argue that requirements *are* not—could not *be*—but in fact, that they *do*, and do *many* different things. The immediate consequence is that requirements need to be researched not only by their form, their expression and their content, but in their specific movements; *in-becoming*, as if it were change, not stability, that should be taken for granted. This implies that for a requirement to be in a world in-becoming it must be studied through space and time. It must be studied through the connections it makes with other assemblages and the various speeds that compose their combinations. For example, a requirement that reaches a high intensity is not born in high intensity, but can make its way there, somehow. Hidden in the requirement’s movements are things in the past, whether technical or cultural needs, and infinite future possibilities that allow the requirement to be a force of change.

That Deleuze and Guattari believe in a world in-becoming and assemblages does not mean they deny the existence, or the interest of efforts to give rest to energies, as long as they do not suppress creative movements that could turn out to be beneficial. What becomes a legal requirement is much more the result of discussions than decisions taken and has another relation to time. The various forces in a requirement's assemblage can overpower each other and lead the requirement to a relative period of rest, having been influenced, and influencing in turn during the period of high intensity. This argument is close to Latour's suggestion to study actor-networks through their points of controversy where their identities come into question (Latour 2005).

There is more than just controversies in points of high intensity. For Deleuze, it is a meeting of primitive forces revealing the potentials of the actors in the various assemblages. The requirement as rhizome is a contraction of time, taking stock of the past, present and future. The contraction carries an influence on the functions that can be applied to it. A highly objectified requirement which makes it materially strong and particularly 'legal' in its state could also hamper further movements and interpretations, limiting its possibilities for further evolving. Paradoxically, the legality of a requirement's state defines vast assemblages of movements illegal. The requirement's future is therefore influenced by the past traces connected to it.

To study requirements as rhizomes, it is necessary to make temporal cuts into the project. Considering requirements as rhizomes implies questioning what requirements are, what are their roles and purposes. It can only be done by apprehending the likely wide and disparate assemblages they connect to and examine the influence they have beyond the objective statement they are often taken to be, lost and forgotten in document.

3.4 Conclusion

This chapter has proposed a theory based on Deleuze and Guattari's philosophy. The theory has an ontology (*in-becoming*); an epistemology (requirements as rhizomes); a language used to make statements and identify how things are produced in such a world (the properties and materiality of the rhizome–multiplicity, assemblage, identity, potential). What it lacks, then, is a function to take the epistemology and use it to make statements about the production of requirements. The investigation as a findings-production machine necessitates a method that does not reduce the potential of the requirements, but that takes advantage of Deleuze and Guattari's concepts to further reveal their potential. To hold a concept, according to Deleuze is to conceive, to create (Smith 2012).

The point, then, is to conceive through their ideas, to develop a function simple enough that, paradoxically, can reveal complex aspects. To do this is to embrace requirements in their multiplicity, something which can only be achieved by discarding Euclidean stability. Instead, what requirements need is to draw a fractal. A simple fractal can be defined (without ever describing it in its entirety) through three steps (Massumi 1992). Starting with a line (openEHR), drawing an equilateral triangle on the previously drawn line (dimensions of requirements), erasing the starting line (abstracting away from openEHR), repeat. The result is an enquiry into the relations between openEHR and requirements, between requirements themselves, and between openEHR, requirements and the world in general. To do this is to instantiate the properties of requirements as rhizomes, to wonder what makes a requirement a multiplicity, its easiness to take so many various forms, etc.; and to ask what machine, what assemblage, what force or power shapes the requirement's properties.

To instantiate the requirement's properties and functions with which it negotiates its shapes is to ask what constitutes the substance and the transformative potential of requirements. The questions raised puts the requirement in the middle, and extends its assemblage to list the forces, the powers, the actors which have a hand in lifting or suppressing its potential to be *in-becoming*, to materialise, as it were, out of its own existence and become something else entirely, something, for example, with another substance and another potential.

- Should requirements be dominated, or should they be seen as allies to the project to enrol other assemblages?
- What are the tools to transform requirements, what are the productions expected of them, what are their influence.

- What goes into a requirement, what forms its substance? How is it connected to?
- What's a requirements materiality? How does it get it?
- How is a requirement connected, how does it engage with the world? What does it mean to be in an assemblage?
- How does openEHR engage with requirements?

To answer these questions is to recenter requirements in the middle, to follow them with their own maps (Latour 2013), following a method that recalls their substance and materiality, examines their past traces and witnesses the hopes of their potential becomings, in the midst of their assemblages. A requirement, then, would not be a statement in void, but an actor with its own materiality (qualities of resistance and accommodation) and potentials. The result is the study of requirements in their own element and under their own terms. A requirement *is* no more, it *does*, it *wants*, it *drives* the identity of the project from one exploration of context to another, it *allies* with processes of work that forego its sequential versioning, it *finds* new routes of expression through partnerships of financially concerned presidents, it *helps* the project maintain its coherence through its accommodation of change and ports.

Table 3.1 presents a summary of the various points made throughout the chapter. It beckons for a different perspective to the study of requirements that does not limit its expressivity from the start, but that lets it travel, map its way through the world and ally itself to all the various becomings it virtually has access to. To do this, the method must follow suit with the theory, not to be its subject, but to confront it in its own becoming. As Deleuze puts it, "[The concepts] must have a coherence among themselves, but that coherence must not come from themselves. They must receive their coherence from elsewhere. This is the secret of empiricism" (Deleuze 2004).

	Singular stability	*in-becoming*
Ontology	Stability is the principal, primordial state. Change is a freakish accident	The world is perpetually *in-becoming*. Stabilising powers exist, but cannot contain desires of entities to explore beyond its own self. Change as normal
Epistemology	Requirements as true descriptors. Requirements reproduce the world along a pre-described order in the requirements DNA	Requirements as rhizomes; transformative entities producing and being produced by assemblages. Causality is rarely evident and flows through several dimensions
Agency	Contradictory agency. Pre-development: no agency, lifeless, overpowered, dumb content. Post-development: true descriptors, complete (assumedly) agency over space (program) and time (schedule)	Complex agency: resistance, accommodation, movement, desires. Complexity derived from transformative potential of requirements
Spatial connectivity	Connected to fictional ideals recreated in symbols	As part of a heterogeneous connection. Connected to assemblages of varied actors making it difficult to define requirements. The extensiveness assemblages reveal how open is requirements sourcing
Temporal connectivity	Pre-existing, post-existing and whole *at the same time*. No transformation in time but through an exact identical function	Contraction of time. Past traces (materials, training, identity, *terroir*) and future potential (transformative futures) are contracted in the present to produce something (e.g. requirements)
Essence	Contained in the statement	Multiplicity of becomings through actions and connections to other entities, spatial and temporal. These requirements are exploring potentials of becomings

Table 3.1: Comparison between singular stability and *in-becoming* requirements paradigms

Part II

Entering the Case Study

Chapter 4

Method

This research started with the necessity to study requirements ‘in the wild’, almost as an anthropological account of beliefs that surround requirements. To do so, and to remain scientifically rigorous in the attempt, it was necessary to use a methodology that would have two qualities. First, it should allow for the observation of requirements in their own terms. This is provided by Deleuzian theory. Second, it should need an environment which would not reproduce mainstream conceptions of requirements engineering, but that would still be understood to be a requirements engineering project. This *terra incognita* was found at the intersection of open source and health IT. Open source remains a contemporary under-researched domain, sufficiently ‘wild’ to be treated without having to refer to traditional notions; health IT added that scientific, rigorous approach that so often escapes accounts on open source. In this way, open source could at the same time be seen as a propitious laboratory to study ‘wild’ requirements, and at the same time, be unlike many accounts made about open source so that it could be generalisable beyond its own frontiers.

This chapter presents an account of the methodology used to analyse the openEHR case study. It is structured as follows:

Looking for a case. How the method was chosen to study the nature of requirements and the implications of interpretivist, qualitative, exploratory research.

Entering the case study. The early phase of data collection. Research context, gaining access to openEHR, and initial interviews,

Coding. The maturing and mature phase of the data collection. Early codes are drafted and put to use on the formal interviews and the lists.

Justifying method. Discussion on the choices made about the use of grounded theory methods.

4.1 Looking for a Case

To research the nature of requirements in the 'wild', it is necessary to understand their natural setting (Benbasat et al. 1987). A qualitative case study is the most appropriate method since it would help theorise concepts that are not identified in the literature. In this sense, the purpose of the method is not to confirm the existence of existing concepts, but to theorise it (Baskerville and Lee 2003; Yin 2003). There are glimpses of such evidence in the literature, but they are often waived aside as 'informal' practices (Noll and Liu 2010; Scacchi 2002). Such theorising calls for a single, in-depth, revelatory case study, instead of a series of comparative, smaller ones (Benbasat et al. 1987). There is a unique quality to open source projects that needs to be appreciated in detail to avoid running the risk of taking short-cuts. Coleman (2012), for example, found deep aesthetic considerations in highly technical open source projects. It is important, then, to consider the culture behind the open source project, lest part of the explanation is trivialised or lacks accuracy. To study requirements in the wild, they needed to be followed (Latour 2005)

As an interpretive study, the context is crucial (Avgerou 2001; Klein and Myers 1999). Interpretivism is often taken as a counter-point to positivist research. Such confrontation comes from attempts by social scientists to emancipate away from the predominance (and perceived shunning) by the natural sciences and their ontological and epistemological positions (Weber 2004). This is a thorny debate, and even historical assumptions, such as the degree of causal explanation, are regularly revisited (Avgerou 2013). Instead of sailing the delicate idiosyncrasies—beyond the scope of this research—an account is given of the principal currents this study founded itself on, regardless of their allegiance to interpretivism or post-modernism (or not), and why they were chosen.

Perhaps, the best way to explain the driver behind this study, and the choice of a case study, is to acknowledge the search for alternative conceptions of reality, often obscured by the large shadows of deterministic measurements. Some interpretive research, beyond the fine points of the composition of reality, admit an interest in the elusive (Law 2004), the relative (Latour 2005), beyond the face-value of unifying meanings (Haraway 1991), in complex assemblages (and production) of actors (Akrich 1992; Pickering and Guzik 2009). There is a minoritarian quality to them, not because they draw the losing side, but because the side is not drawn, or difficult to draw; how else could there be so many multiplicities? When Law (2004) reflects on the 'otherness' of alcohol advice centers, what difference is there from the intangible quality found in the creation of user-led innovations (Urban and Hippel von 1988)? The driver, thus, was not the study of the pitiful, or the oppressed, but of alternate realities that would better account for complexity and messiness than the deterministic relations that are mainstream.

The choice of a case study settled firmly on openEHR after a small pilot interview and a limited period (two months of perusing on the mailing list) of archival analysis were carried out.[1] openEHR showed a level of complexity unlike many open source projects. This complexity was its immediate attraction because openEHR had dealings between varied domains: health, informatics and open source. From its creation, it belied a Quixotic attempt at changing healthcare, yet remained lucidly on the ground, weighing issues that more technical open source projects would not reflect on. Health, for example, is a domain agitated by contradictory cultures: the open nature of clinicians' approach to knowledge, and the closed systems many of them work with. It was clear, early on, that open source was to

[1]The protocol setting up the initial pilot interview is reproduced in appendix A. This pilot interview was also used to judge the access level that could be provided by openEHR.

be a controversial matter in openEHR, not in itself, but because of the multiple understandings and application that open source provided to define the project and its requirements. openEHR, since its inception, was a controversy aimed at destabilising the status-quo and provide new avenues of research and application. openEHR seemed to appeal to the theory's political inclinations of movement and independence from institutional norms. This conforms with Avgerou when she cites Weber: "A major aspect of identifying and delineating a research object in IS research is the theory about society and technology adopted by the researcher. Such theory guides, either explicitly or implicitly, the choice of some focal entities rather than others, the meanings given to them, the associations, and the constructs studied by the research (Weber 2012)" (Avgerou 2013). As such, studying the nature of requirements can be messy: it implies exploring their contours, theorising their existence; not explaining clearly identifiable, natural objects.

Such messiness is suited to an exploratory, single-case study because the case is both revelatory (for requirements), and unique (for open source). It is revelatory for requirements because their nature is not usually discussed, and unique because the case, openEHR, is not the typical, technical, open source project. Often, such projects respond to particularly individual drivers to innovation, often being the need to "scratch an itch" (Hippel von 2007; Raymond 2001). openEHR, in contrast,had a careful consideration of the importance of requirements, and understood them to be abstract things which potential needed to be articulated—they saw a middle to requirements that needed to be negotiated, an in-between the concrete and the overly general. In that sense, openEHR was, since its inception, explicitly a requirements project that could reopen the debate concerning their nature.

4.2 Entering the Field

This section principally refers to the early and maturing phases of the research, which correspond to step 1 and 2 of the data analysis (c.f. tables 4.1 and 4.4. The principal methods for investigating openEHR as a case study were through two means: analysis of the mailing lists and interviews. Analysing the mailing list is a frequent research method in open source studies, while less applied in software engineering (with some notable exceptions (Laurent and Cleland-Huang 2009)). Interviews are common features of information systems research (Davidson 2002; Urquhart 2001). These previous two cited studies use interviews to understand how requirements are developed and the various human and technological roles that shape them. This is fine to study requirements as objects, but not as their own actors, which the theory, and STS studies in general believe ontologically. Using mailing lists as the main source of data for studying the nature of requirements proved essential in establishing requirements as dynamic actors. Considering requirements in their own environment showed the extensive assemblages they could draw upon and how they could alter realities to their convenience. Nonetheless, interviews were a useful first contact to the realities of openEHR and the issues it faced.

One of the advantages of studying open source is the available information prior to starting research. Exploring the archival records of multiple sourced documents about openEHR was an opportunity to get acquainted with the language and themes that the participants were concerned with. White papers from openEHR, literature on health EHRs and openEHR itself were the first access points to the project. This helped in setting early questions, identifying key people to interview, and writing a pilot protocol to give to the project members who were interviewed.[2] Not to be dismissed, being able to 'speak' openEHR afforded an easier time with the interviewees and a certain level of complicity that would have been impossible otherwise. This was useful also for following up on questions during the interviews—mentioning specific events for example. openEHR proved to be remarkably open about their work, reflecting into what they saw were failures or successes.

Open source research is often based on analysing the mailing lists or visiting the project's IRC channel. Seldom are open source projects participants interviewed, perhaps because of the readily available data. Archival data provides valuable insights into the everyday life of a project, but risks missing deeper, richer reflections

[2]See appendix A

on the nature of open source (Shaikh and Cornford 2012). Interview data provides a particular understanding that can be fruitfully combined with archival data. For this project, data collection took place over different phases. In the early phase, data was collected ad-hoc and informal interviews were carried out (3) prior to the formal interview round. This helped to have an early understanding of the principal controversies, a better idea on the context of study, and a context to the mailing lists which were to become the primary source of data for this research. Table 4.1 presents the three principal phases of data collection, which corresponds to step 1 in table 4.4.

Phase	Data collection	Time
Early	Archival exploration, conference assistance, informal interviews	2009 – 2010
Maturing	Semi-structured interviews, tool-assisted codification of interviews and mailing lists	2011 – 2012
Mature	Evaluation of codes and re-codification	2013

Table 4.1: Data collection phases

The process of data collection was inspired by Latour's exhortation to follow the actors and letting them speak (Latour 2005). Deleuze makes much the same call when he says: "It's not a matter of bringing all sorts of things together under a single concept, but rather of relating each concept to variables that explain its mutations" (Deleuze 2003). This allowed the analysis to be centered around particular movements, as called for by Callon: "to understand the functioning of organisations, we have no choice but to explore the role and the effects of the varied and evolving organisational instruments" (Callon 2002). So, methodologically, it is not only the actors that should be followed, but the actors through their transformation, or as Guattari would say, the production of their subjectification (Guattari 1995). There is a performative sense to the method that echoes the theory's concern for the production of realities. It is both out of respect to the actors' potential for change and to give an accurate account about the forces in play in larger assemblages. This was an essential prerequisite to understand the full reach that requirements had over the case study and the field in general.

Following actors and letting them speak might sound as if the researcher had hid behind curtains, a ploy to convey an air of neutrality. In some sense, there is a possible contradiction in both following the actors *and* letting them speak since following them depends partially on which actors the researcher chooses to follow. Although shared with other types of research, this difficulty is particularly marked in interpretivist research precisely because the researcher is seen as *in* the world. The researcher's own interests, or the actor's capacity to grab his attention, are both influential to and influencing on the 'objectivity' of the research. This is a political turn on the other types of research which hold closer to the language and mechanisms set by the natural sciences (Lehman 2011), though perhaps less so than critical studies (Myers and Klein 2011). The researcher's role is not neutral and probably could never be entirely so (Law 2004). The researcher, even if unwilling, is necessarily drawn to his own interests, but not—and this is where positivist critiques of interpretivist or critical studies err—limited to the researchers' subjectivity. In other words, the researcher's subjective experience necessarily involves a "'swerve' [*déportement*] of the self that does not leave him as he was before" (Zourabichvili 2012). There is no intentionality as such, but rather directionality.

4.2.1 Interviews

The directionality for this project was inclusive. The informal interviews helped in choosing the sources for the more formal, semi-structured interviews. The informal interviews helped in putting context to openEHR, but also health IT in general, which were both a difficult topic and domain. The informal interviews usually took place at a conference where I took advantage of the opportunity. These were not recorded, but the informal nature permitted a certain candour and personal implication that, at the beginning, was not clear that they could be achieved in more formal conditions. They were also useful to confirm personally the extent of understanding of openEHR. Following the informal interviews, the interviewees were asked if they would consider formal interviews, which they all accepted. This is step 1 in table 4.4.

In the maturing phase (c.f. table 4.3 and step 2 in table 4.4), the choice of sources was as varied as possible to allow an account as full as possible about the different views on requirements. The list of interviewees included technical (2), clinical (2), foundation (2), and people sitting on the fringe of the project (2). The participants were all active in openEHR, but differently so. Some were trying to advance openEHR's political agenda in accordance to their country's national interests, not owing any formal loyalty to openEHR but their own belief in its potential, others were in more proselytising positions (c.f. table 4.3). When formally interviewing, the mailing lists had only been explored informally, but had already derived some of the important and controverted issues related to requirements.

The formal interviews were semi-structured and regulated by a protocol, which was sent to the interviewees prior to the talk. They were given a paper version in person, if possible. The protocol established the research purpose, the goals of the interview, and a basic set of questions. The interviewees were also informed that they could stop the interview whenever they wanted, and whether they felt comfortable if the session was recorded. The list of questions for semi-structured interviews was tailored every time, apart from a basic introductory set, such as their opinion on the state of openEHR or what were their motivations. These were useful to break the ice. Depending on who they were, specific themes were developed for each interviewee and tended to correspond to two types: questions relating to their particular implication and background (engineer, clinician, etc.); and more general inquiries that had been identified on a previous interview. Interviewee names and opinions were avoided in discussions, unless it had been part of a publicly available debate on the openEHR mailing lists. Every interviewee was asked if they could be recorded prior to the interview, and notes were taken highlighting themes and items of interest. The interviews were transcribed as soon as possible to avoid losing the 'situatedness'.

The location of the interview changed often, depending on the availability, and where they were. The semi-structured interviews were conducted online though voice over IP (3), through IRC (2), others at the interviewees' office (2), a hospital (1), or even the pub (1). The formal interviews followed a simple protocol whereby I would introduce myself and the research topic and briefly explain my interest with openEHR. At the end of each interview, the thesis was put into question: 'How does this change the established notion of openEHR?, 'How does this differ to typical requirements engineering?' The point was to have an incremental appreciation of the difficulties of openEHR, and how these related to requirements. Themes began to grow out of the discussions, which could then lead to discussions beyond the protocol. So, despite the starting formality, once the protocol was satisfied and the important, planned questions answered by the interviewees, discussions took an air of informality. One interview, in particular, had to be continued in a nearby pub because of a fire alarm being triggered.

What the interviewees talked about matched their 'public' identity on the mailing lists, which amounted to great satisfaction to know that the mailing lists and the interviews validated each other. It was also pleasant to see that openEHR had not suppressed any of its participants to follow an official position, but instead responded as they wanted to without any pressure.

4.2.2 Mailing Lists

Despite being an open source project, and much research basing itself on archival exploration, the mailing list was not such an obvious starting point. Many of the lists (the technical list chief among them) were, for some time, problematic to understand because of the specialised discussions taking place there. The interviews were helpful in that regard, since they offered some context and clarifications on the issues that were discussed there. This is why, formal analysis of the mailing lists took place after the start of the semi-structured interviews. As time passed, it was easier to understand most points that were made and to get a better handle of the finer details of discussion. Most notable were some recurring issues that reappeared regularly in several mailing lists. These were helpful in trying to identify the various debates going on. These debates were sometimes used in the formal interviews to elicit more details. This corresponds to the early phase of the research in table 4.1 and step 1 in table 4.4.

The announcement and technical lists were first coded informally, using tables and summaries which proved unwieldy to manage. Going back to the source was cumbersome, and often relied on memory, which made it difficult if any amount of time had passed between the coding and the summarising. After some frustrating restarts, a more systematic approach was taken and used an open source programme for coding called TAMS (Weinstein 2006).[3] The main reasons to use TAMS was that it ran over all principal operating systems, and therefore could be used anywhere. TAMS was used to code the mailing lists, as well as the semi-structured interviews. The contextual information gained from step 1 and the formal interviews, and the use of a tool helped this research enter into the maturing phase (c.f. table 4.1) and step 2 of the data analysis (c.f. table 4.4).

Three principal mailing lists formed part of the systematic coding process: the technical, clinical, and announcement lists (c.f. table 4.2). These lists were more promising for several reasons: first, they were more general than other lists (there are various lists on the implementation of openEHR in Java, or Eiffel); second, They were the most populated lists in terms of emails. The range of time used to limit the data collection size was set to the period from 2009 to 2010, the start of this research. The other reason for this range was to study contemporary events, but if these were too recent, their importance could go unnoticed since they would not have had time to leave a trail in the lists. Despite the time limit, the size of the evidence that needed was quickly overwhelming. The technical mailing list on 2009 alone counts for more than 1376 emails. Put together, the three lists amount to approximately 2113 emails (c.f. table 4.3).

Both the emails and the interview transcripts were coded through TAMS, with the application providing the list of codes on the side of the source material. With this new way to code, the 2009 technical list was coded in its entirety. The process was long, and even when using TAMS, the codes were re-evaluated again in 2012

[3] Text Analysis Mark-up System

(c.f. table 4.5, and step 2 in table 4.4 for the initial codes; and 4.6 and step 3 in table 4.4 for the revised codes).[4] This prompted another major recoding of the lists. The last coding phase went through several starts and stops, but when the scheme seemed solid, a clean coding was carried out in this order: the transcribed interviews, the technical list, the clinical list, the announcement list. During this last phase, theoretical saturation was reached towards the end of the technical list's recoding, and could have proved a good moment to stop (Charmaz 2006; Urquhart 2012). However, the use of several lists complicated this decision. What if important data became apparent in the clinical list? Despite the redundancy, the confirmatory nature of the data in the clinical and announcement lists was helpful to validate the emergent model.

List name	Content
Technical	Detailed discussions on technological aspects of requirements and technical artefacts. Prone to some over-spilled into less technical discussions
Clinical	Detailed discussions on clinical concepts and requirements
Announcement	Official list for Foundation announcements. Referenced from the technical and clinical lists to know the official position on controverted issues

Table 4.2: Coded mailing lists

4.3 Coding

This section presents a detailed account of the elaboration of codes and their presentation in the thesis. It corresponds to steps 2 and 3 in table 4.4, which presents a synthesis of the data analysis process; and the move between the maturing and the mature phase (c.f. table 4.1).

4.3.1 Practical Coding Issues

There were several practical concerns of note relevant in understanding the way the coding developed:

[4]The revision of codes, and the emergence of themes is discussed in section 4.3

Source	Events	Quantity
Mailing lists	Technical	1376
	Clinical	698
	Announcement	39
		Total: 2113
		Word count: 227500
Interviews	Informal • Clinical core (1) • Foundation core (1) • Satellite project founder (1) Formal • Clinical core (2) • Foundation core (2) • Technical core (3) • Brazil national health agency (1) • Satellite project founder (1)	3 informal interviews 9 formal interviews 7 respondents Total: 12 interviews Word count: 65000
Participant observation	Board meeting (1)	1h

Table 4.3: Data Collection

Early codes were highly variable. The first coding items were created first from theory, and complemented from the data gathering. Interviews were prominent to setting up some important codes such as what would later become the 'identity' theme. As the codes became more systematic, and the theory and ideas more coherent, they were put into a coding scheme (c.f. table 4.5). This went through some substantial changes while further coding, specifically when coding the 2010 technical mailing list. The last change to codes were to include requirements qualities (both about the process inducing capability and the requirements themselves). The codes thus came both from theory (researcher codes) and openEHR (field codes). Towards the end of the study, it became hard to distinguish between researcher codes and field codes, except by the historical name given to the code.

Link to source. Summaries, without the support of TAMS, were problematic because of the loss of link to the original source. With TAMS, summaries took another shape and resembled closely what is called 'memos' in grounded theory literature (Urquhart and Fernandez 2013). Memos highlighted the recurrent, highly controversial, or interestingly mundane events taking place.

Steps	Tasks	Output
1. Informal coding	• Categorise emails from announcement list • Exploration of the technical list • Contextualise the project through informal interviews	• Derive main issues of project according to official channels • A few codes stand out. No hierarchy or clear relation
2. Formal coding	• Formal interviews • Establish list of codes from theory and grounded codes from informal coding • Start coding technical list and formally code announcement list and interviews	• 19 selective codes, 57 open codes. Some hierarchy, but unclear relations • Better understanding of contentious issues
3. Revising codes	• Identify key themes from the codes (involved multiple starts and stops while coding technical list) • Review relation between emerging codes and theory	• Revised coding scheme (172 codes, 3 principal themes—c.f. tables 4.6, 4.7) • Clearer relations between codes • Emergence of grounded model relating the themes • Evaluation of codes and theory • Recode all data

Table 4.4: Data Analysis

The evidence was often corroborated by its existence in the other lists and interviews.

Different source qualities. Interviews were qualitatively different from the emails in the list. Though contributing to what Yin (2003) calls evidence triangulation, it felt strange to code data so seemingly unlike. Interviews were more reflective in nature, while the emails in the lists had both more 'spur of the moment' reactions, and reflective summaries of the various points made by the participants. Though the points they made converged towards the same themes, for some time, it was not clear whether they should be treated alike. Being able to corroborate what the interviewees said in the interviews from their positions in the mailing lists helped assuage the uneasiness.

Controversies and concerns. Through coding, the existence of two type of 'movements' seemed to emerge: the controversies (in the Latourian sense of a transformative drive), and concerns (a milder, less intense, but apparently more latent desire for change). The controversies were found to be largely contained to similar threads of emails (although the occasional spills were observed), while latent concerns were present in many different threads (although the occasional high concentration was observed and became part of a controversy). As coding went on, it became clear that the latent concerns eventually were contextualised by controversies.

Selecting codes and themes. The threads could not be selected prior to coding because it could not be known in advance which would be interesting; the coding was thus systematic to all threads, though the emails that were clearly unrelated to requirements were ignored (e.g. server downtime announcements). In order to have a higher view of what were the main controversies, for the participants, a small Python script was written which simply counted the frequency of appearance of subjects in the lists (c.f. appendix B). The results from the script were used to confirm that the principal controversies (in relative terms of emails) had been analysed.

Coding threads. Coding threaded emails presented the difficulty of contextualising data. Sometimes, emails were only a few lines, at other times they were essays with in-line replies to several emails. The meaning of an email was thus relative to the other emails in the thread. A few of these threads even spanned several lists, which needed a careful reconstruction of the argument. The meaning of controversies was often found in the whole of the thread, and not necessarily in particular emails. On top of this difficulty were the sequential presentation of the emails from the monthly mailing list data dumps. These were ordered by a time stamp and the contextualisation of the intent of the participants required particular attention to whom the reply was addressed (found following his email). In this way, the thread could be reconstructed on its whole. Sometimes, though, some emails were cut by the server, and could only be found in further participant replies. It is this unordered state of emails, which, even if essential to rebuild participant's arguments, made the size of the data particularly large. The estimated total word count of the three mailing lists during 2009 – 2010 from the server's data dumps amounts to 227500 words (including headers to know who sent the email and the chain of replies).[5] Memos were useful in this process of reconstruction: themes could be related to various threads, which helped identify those which stood out. Some of these were put inside a table (c.f. appendix C).

What data to present? Selecting which particular emails among the large quantity of evidence proved difficult. The selection could risk losing finer details but was necessary for a clear presentation of evidence. This was particularly tricky since the threaded emails contained an overall rational and context which should not be discarded from the presentation. On top of that, and given the vast size of the data, only a few emails could be presented, and even these would have to be cut. The solution chosen was to introduce the context in the analysis, presenting it in relation to the various selected portions of emails. This way, parts of the context could be in the background, while not overloading the analysis argument with large and cumbersome data sizes.

Entangled codes. The analysis of the coded email threads led to a surprise: codes were often close to each other. Even on the technical lists, several issues concerning requirements would be entangled together. This eventually gave rise to closely linked themes. There is a good reason for the intermingling of requirements functions: the close relation of codes in the data paragraphs showed the potential that requirements held and the assemblages to which they connected.

4.3.2 Formal Open and Selective Coding

Formal open coding started in the maturing phase (c.f. table 4.1), after initial informal interviews (3) and previous informal coding attempts in the early phase

[5] This estimation was done through the correspondence of kilobytes to words (1 to 140). The size of all the monthly data dumps were added and subsequently transformed into word count.

of the data collection. The maturing phase, at the time, was not supposed to be initial, but was intended to form the codes that would define the analysis and discussion for this research. This early coding scheme was drafted before formally coding the mailing lists and derived from the semi-structured interviews, and the early takes on theory. The codes were used on the formal interviews and half of the technical list. The main constructs that came out clearly from the already gathered data told a story of balancing control, stability and drifting (c.f. table 4.5). These stories (from the selective codes) lacked perspective, and it was not clear, for example, what was emerging, or the role of requirements in that emergence. The open codes were too disconnected from the selective codes, and not contextualised sufficiently. Endemic to the codes' problems was a continuous learning of the theoretical basis of this research, which made their understanding tenuous at the beginning.

What was clear, though, was the tension between the informality of requirements (almost necessary to cope with change), and the need to show legitimate, 'strong' requirements. The problem, though, was the relation between the open codes and the selective codes, which was not evident that one necessarily emerged from the other. This led to a re-evaluation of the codes, with a focus on formalising the relation between open codes, and selective codes. To do this, it was necessary to recontextualise requirements and think of their relation to the project's concerns.

4.3.3 Revising Coding

The lack of contextualisation of the codes and the murky relations showed that codes were symptomatic of both being overly descriptive and overly general. The 'rhizome' and 'becoming' code, for example, were a misuse of the theory on the data. Rhizomatic requirements are in-becoming, and thus was assigning an ontological quality directly to the data. In contrast, the open code 'assemblage' was overly descriptive. It muted some of the ways in which assemblage (or the possible lack thereof) could exist. This code suppressed the more subtle ways in which assemblages could behave by making them be 'destabilising', 'stabilising' or

selective codes	open codes
assemblages	actors, destabilisers, maintainers, stabilisers
becoming	change, controversy, desire, explosion, illogical, loss, rhizomatic, standards, unexpected, unsettling
bricolage	n/a
business model	n/a
education	n/a
emergence	change, improvisation, innovation
emotion	hospitality
engagement	collaboration, doers, feedback, innovation, loss, protection, rigour, voluntary
goals	foundation
projection	n/a
quotes	n/a
requirement process	n/a
requirements	alliances, collaboration, coordination, criteria, engineering, exclusive, historicity, local, negotiation, selection, sourcing
rhizome	n/a
rigour	best practices, constraints, standards, durable, loss
speed	n/a
territorialisation	n/a
time	n/a
values	authority, clinical, constitution, democracy, freedom, loss, no risk, openness, ownership, practical, technical, trust

Table 4.5: Initial open and selective codes

'maintainers'.

What later evolved into the themes were already present in these early codes. For example, there was a sense of protection of certain actors in particular ('protection', 'doers'), or a necessity to both hold requirements down and let them emerge ('rigour', 'constraints', 'engagement', 'emergence'). What was not clear was the 'why' these codes were important to the participants, and how theory could make sense of them. This 'why?' led to a more concrete question which precipitated the revision of codes: 'why and how are requirements important to openEHR?' This question, in particular, helped contextualising requirements and make sense of the relation between the codes. Taking a step back and going through the old codes and the source material together, several controversies and concerns became apparent:

- openEHR was defining what it was, *who* it was, and what it could do
- openEHR was careful in what it was proposing and took great pains to

document and anchor their requirements

- openEHR was projecting itself in relation to competition, satisfaction of requirements, how the health domain was moving
- openEHR was worried about temporal and spatial issues (sustaining innovation, the speed and time to engage requirements, openEHR's actions and external influences on the core, the openEHR movements in Australia and Brazil which started their own CKMs—where to engage?)

New associations between codes, both old and new grew from these concerns. Several concerns stood out: its position in the world and what people thought about the project; the territories where openEHR was being implemented or could introduce itself in; the possible ways to manage requirements; the speed that it took to respond to change; and the values and qualities that should be respected. openEHR, for example, was aware of its minoritarian position in the world and on the lookout for possible partnerships that would increase its public profile. These concerns, grounded in the data, became a meeting point with the theory which would put these new associations, and the theory itself into perspective.

With this in mind, the new coding scheme could be built while coding, associating the phrase and paragraph as an open code to one or several of these concerns. The resulting open codes and selective codes are shown in table 4.6.

4.3.4 Theme Building

With the set of growing selective and open codes on one hand, and openEHR's concerns on the other, it was time to rethink their association, their wider meaning, and their relation to requirements. For example, why did it matter to openEHR to be part of the world? The open codes pointed that they were trying to 'build a community', 'protecting the core' such as the 'doers' (certain actors that acted instead of talking). These two selective code were about openEHR's concern to be seen in the world, but also about what kind of project it should be: it was an identity issue. In some sense, there was a paradox to solve: create a larger community (and have new requirements), yet protect what is already there (maintain the requirements and the original assumptions).

openEHR. especially the official channels, were concerned about where the project was physically. Implementation, that anchor to realities, was a particular concern, but did not appear as such most of the time. The concern was to be able to bind requirements to a place, a context, a rational. For this, openEHR's requirements needed to be exportable, diffused. It should be easy for someone to contextualise openEHR's requirements to their particular problem. In part, this was the attractiveness of open source, but how to do it without hurting openEHR's ability to market itself as the source of clinical requirements? Thus, this territorial concern was about openEHR's ability to deploy its requirements in its own terms.

openEHR's requirements were central to these concerns of identity and territory. Part of its minoritarian position was its rejection of committees (often seen as political by many participants) in the drafting of requirements. Others were more extreme in this concern and believed openEHR's way of engaging requirements should be controlled at the ground-level, openEHR functioning more as a guiding hand. How much should openEHR's requirements be emergent, and how much should they be stabilised? This was not a binary concern, it touched on the concept of a requirement and its capacity to represent reality, and which one. For some, it would be a place holder of descriptive measures (a generalisable property-actor), for others, an actor that could shape clinical realities, by its presence or its absence. The role of values had an indirect incidence on requirements, but still informed the process.

These three themes were directly linked together, forming part of the requirements' assemblages. The theory guided their relation, explaining the links between identity, territory and engaging requirements. The concept of becoming could make sense of how openEHR saw requirements as open ended promises that needed to be articulated in different ways. Deleuze and Guattari's concepts could explain the complexity of requirements through movements of space-time that traditional requirements engineering would have simplified away. Values and movements were indirect themes that supported or influenced the three principal themes. Their roles were supportive to the participant's arguments and concerns.

The principal themes, having been identified through their relation to selective and open coding, and related together by theory, are shown in table 4.7.

4.4 Justifying Choices

A core tenet of grounded theory is to approach data unburdened by preconceptions from theory or literature. This is a point of contention between proponents and detractors. This has some consequences for those researchers who wish to use grounded theory for its methods, but have settled on a theory. This section describes how this research looks for a compromise between the demands of grounded theory and grounded methods of coding, while using a theory not directly derived from case data.

4.4.1 Coding Decisions

It is disputable whether this research follows the core tenets of grounded theory methods. Above all, it is the codes' mixed parentage of theory and empiricism which might not be well received by 'purists'. For theories to be grounded, practitioners advise a minimal level of baggage to avoid carrying preconceptions into when collecting data, but recommend against having a 'blank slate' instead of a researcher (Urquhart and Fernandez 2013). The problem, especially for those researchers looking for a middle ground between grounded methods and other theories, is negotiating what that 'blank' state is.[6] Such line drawing can be profitable by improving rigour in qualitative research, but it can also be misunderstood as problematic de-legitimising of other theories.

This study, from the outset, questioned assumptions on the nature of requirements. It took a minoritarian view that, perhaps, the way requirements are conceptualised and seen by the mainstream literature is too narrow. This formed the

[6]Does this make non-grounded theory ungrounded?

so called ‘literature gap’. In some sense, this research carried a double preconception: the existence of a mainstream nature of requirements, and the possible existence of an alternative. To what extent could this be problematic for the validity of this research? Deleuze, and STS studies in general, bear in mind the various realities produced by the actors.[7] Latour’s most recent work focuses on the multiplicity of modes of existence—actor-‘ontologies’, so to speak (Latour 2013). In all these foreign worlds, the researchers’ prerogative is to be his own person, to trace cartographies of the actors’ realities. The researcher is active, but careful.

Glaser and Strauss (1967, p.253) do not seem far from this position: “*Our* point is that his principal insights were based on his personal experience as a cabbie. Some experience that formed the basis of his later systematic theorising undoubtedly occurred while he was still a cabdriver, and others—perhaps the major ones—occurred later when he reviewed his earlier experiences.” There is a knowledge baggage here as well, and a personal one too. There is a reason why research is undertaken, some paradox that has been identified from which the research unfolds (Harman 2009). Grounded theory, though, is looking for more than ‘having’ past experience or knowledge of the field, but looks to “*deal* [emphasis added] with pre-existing knowledge bias and a way of integrating this knowledge with empirical data” (Urquhart and Fernandez 2013, p.226). How to deal, indeed, with this contamination problem the researcher is a vector of? To answer such a question directly would be to assume that data themselves are neutral, uncritical, vulnerable to the only contagious factor of research. In other words, two condition must be met: the researcher is contaminating, *and* the data are neutral so that they may become contaminated.

Deleuze and a large part of STS studies are concerned with the *articulation* of data. Following the actors, as Latour (2005) exhorts, is not neutral. The researcher chooses the story and how to pursue it; there is an agenda to find *something*. Grounded theory proposes that such ‘something’ not be tainted through the appli-

[7] Actually, Deleuze does not see theorising work as ontological, but rather would describe it as empiricist, necessarily grounded, and stemming from critical thinking (*pensée critique*) (Deleuze 2004; Zourabichvili 2006).

cation of some methods that require relating codes together, but that relation is done by the researcher. What would compel the researcher to avoid involuntarily tainting that data? Deleuze refers to this as an attitude: to *pre*hens; that is, the world should be experienced previous the forming of suppositions (Deleuze 2006). This is a noble goal, and this research has tried to follow the advice as much as possible, but it remains to be seen whether methods can, as a general rule, control behaviours, however involuntary as they may be (or precisely because they could be involuntary).

If the researcher is the contaminator—and should try to contaminate the least possible—it remains to be seen whether data are at all neutral, but yet needs the constructing hand of the researcher to build into a theory (Biernacki 2012). How does this affect the elusive, the tacit, or the powerless? Is this not a utilitarian perspective that would assume all data are born equal? In this sense, it might befall the researcher to dig that subsumed data up, to present it alongside the stronger data. Data, just as the researcher, carry their own productions, prejudices and beliefs. Can there ever be enough data to tell the whole story?

Despite these points of contention, primarily contesting the efficacy of grounding theories, the methods they provide are useful to build models that can act as a middle-ground to theory (Urquhart 2012). The rigorous and systematic approach is useful when the abundance of data is such that there seems to be no grasp to the theory's application. The most helpful aspect were the multiple coding stages and the reflective pauses after each one, thereby treated data as relations instead of separate ideas that needed to be picked, thereby making a chain of evidence (Yin 2003). This is also a result of the detailed coding strategies that, because they emphasise the minute details (Charmaz 2006), are helpful to treat data as a whole, contributing to the emergence of a model about the phenomena and their relation to Deleuzian theory. Above all, perhaps, it is useful to the novice researcher who is unsure about how to treat data.

4.4.2 Theoretical Sampling and Construct Validity

Theoretical sampling and construct validity are conditions that Yin and others put on the solidity of case study research (Yin 2003). The various data sources differ in many respects:

- interviews are self-reflecting exercises about past events and personal projections of what openEHR should be in the future
- informal interviews were very similar to semi-formal interviews because of the willingness to be open by openEHR's participants. It could not, in the beginning, be necessarily counted on
- semi-formal IRC interviews, even if they took place once because of the participant's difficulties in oral English, had nonetheless a different substance than the oral semi-formal interviews
- mailing lists are 'in the moment' discussions that range from technical discussions to high-level, philosophical questioning of openEHR's role in solving EHR problems
- mailing lists varied in subject and tone. The announcement list consisted of board messages that reproduced and justified the board's decisions and goals. The technical list was varied and often sought to understand the implications of technical requirements in the wider context of processes, values and identities. The clinical list was concerned about the sourcing and articulation of requirements from a clinical perspective
- white papers had a summarising goal and a reproduction of the core's group values and thoughts

This variety of construct quality is an argument in favour of the solidity of the case study. This variety in quality validates the consistency of the issues and their negotiation by placing them in different temporal and spatial contexts: temporal because participation took place along various project lifetimes and frequencies;

and spatial because they were discussed in different means and physical locations. Temporal and physical variety is a useful indication that the controversies and concerns existed outside artificial boundaries that could have been intentionally or unintentionally created by the researcher or openEHR. The various spatial-temporal combinations created markedly different contexts from which to make Deleuzian ‘cuts’ along movements of interest (Deleuze and Guattari 1987).

4.5 Conclusion

This chapter has first presented an account of the reasons and justification for choosing openEHR as a case study, and the methodological instruments that would assist in its analysis. The search for a case responded to two principal necessities: an escape from mainstream accounts of the nature of requirements, and a context rich enough to allow for a temporal study. In a way, this research tried to ‘dislocate’ requirements away from their comfort zone. To this end, an interpretive analysis is fruitful in considering what could a requirement be. openEHR promised a unique setting to look for alternative explanations on requirements since it is both an open source project and evolved in the complex environment of health IT. openEHR had had, since its beginnings, a careful appreciation of requirements that promised to be revelatory.

To study the case, as other open source account, emails from mailing lists were collected and codified. These were also complemented with interviews of core participants who put helped put a context behind openEHR. The emails and interviews were coded using methods from grounded theory. No claim is made here about an emerging theory. Still, a compromise had to be negotiated between grounded theories’ recommendation to carry a minimal baggage of literature when undertaking the analysis, and the choice to study requirements in a new light using Deleuzian concepts.

In the next chapter, openEHR and its context are presented.

Selective codes	Open codes
being relevant	independence, aspiring to goals, becoming majoritarian
protecting the core	being coherent, protecting doers, educating, protecting doers, protecting foundation, users, risk aversion
being represented	making alliances, being publicised, showcasing potential,
building a collective	internalising actors, attracting contributors, building an open source community, making alliances, aligning with the clinical community, aligning with the technical community, educating
open	open source, democracy, sharing, including, freedom, honesty, sharing, transparency,
closed	dictatorship, facts, excluding, framing boundaries ownership
authority	single reality, facts, personal, control
practical	engineering criteria, in practice, in reality
technical	technical criteria, scientific, purposeful
trust	inviting discussion, welcoming contributions, allowing
clinical	clinical criteria
foundation	openEHR goals, personification, core, constitution
recontextualising worlds	expanding problem domain, introducing new contexts, challenging the core, localising context, explore problems
decontextualising worlds	narrowing scope, generalising, categorising problem domain, render explicit
diffusing openEHR	spreading their identity, exporting openEHR, innovating, externalising efforts, business model
reproducing	relating to competitors, relating to as-is, imitating, aligning with current realities
qualities	ambiguous, anchored, cascade, clean, clear, competition, complex, consistent, context specific, core, crystallised, definitive, dense, fragile, frequency, future proof, generic/contextualisable, guide, interoperable, invisible, levels, monstrous, multiple, multiplicity, nature, non-destructive, pliable, precise, re-opened, reality, relations, responsibility, rigour, safe, shapeable, simple, singular, strict, ubiquitous, uncertain, useful, visible
stabilise by rooting	rigour, control, formalise, standards, implementation, historicity, material
emergence by becoming	unexpected, challenging the core, uncontrolled, bricolage, projecting use cases, evaluating practice, recounting personal experience, scratching an itch, opportunistic, organic/raw, ungraspable/wild
coordination	judging a good requirement, legitimising a requirement, arguing for a requirement, emotion, doers, management, committee
strategies	balancing act, internalising requirements, partnering, perspectives
movement	speed, time, space

Table 4.6: Revised open and selective codes

Themes	Selective codes
Identity	being relevant, being represented, building a collective, protecting the core,
Values	authority, clinical, closed, foundation, open, practical, technical, trust
Territories	decontextualsing, recontextualising, reproducing, diffusing
Modes of engagement	emergence, stabilise, coordination, qualities, strategies
Movements	speed, time, space

Table 4.7: Relation between themes and codes

Chapter 5

Case Study

The concern for information systems in health services is not new: Nightingale already complained about the challenges they posed in the 19th century (Ingram and Arikan 2013). Even the introduction of seemingly unproblematic information systems such as checklists has been difficult, despite their widespread use in other safety-critical domains (Gawande 2011). Recent efforts in the UK to adopt new health information systems have also been met with considerable criticism (Clegg and Shepherd 2007). In 2002, the NHS launched the NPfIT programme which intended to revolutionise the UK health service through the adoption of IT systems (Takian and Cornford 2012). The project was immense: 330 health institutions in England were to take part (Coiera 2007), with an estimated budget of over £12bn over 10 years (Coiera 2007; Hendy et al. 2007; Takian, Petrakaki, et al. 2012). Despite some sizeable achievements, the programme was unsuccessful (Greenhalgh et al. 2010; Robertson, Cornford, et al. 2012).

Introducing information systems in the health service is a particularly difficult task. First, how the information should be used is difficult to establish. Should it all be made available to the patient so that they may decide what they think is best? Should medical staff use their experience to filter the information and walk patients in their decisions? Roles and conditions can be unclear (Petrakaki et al. 2011), only sometimes empowering patients (Agarwal, Anderson, et al. 2013; Kuijpers et al. 2013). The mixing of ancient concepts such as the Hippocratic oath with contemporary values of freedom, choice and care (Batifoulier et al. 2012; Mol 2008) results in a strange domain. Moreover, not only must health information systems meet stringent safety requirements (Barber et al. 2011; Coiera et al. 2012; Hayrinen et al. 2008), but health services must also integrate patient safety at all levels (Lewis 2005).

Second, information systems are not neutral objects: their conceptualisation of medicine and healthcare embodies certain political and cultural thoughts and decisions (Sjögren and Helgesson 2007). For example, certain professions, such as nursing, tend to be disregarded despite their central role (Andrew 2003). Unique institutional contexts prevail in the health domain (Chiasson and Davidson 2004; Ham et al. 2003), making comparison to other domains difficult.

Third, there doesn't seem to be a single right way of developing health systems. In arguments that echo software engineering discussions that are still ongoing, different countries adopt information systems differently: some through top-down, others through bottom-up efforts (Coiera 2009). Ultimately the debate is about creating useful systems that cater to local needs but are also interoperable at larger levels (Hendy et al. 2007). Much of the criticism thrown at the NPfIT had to do with these processes. The NPfIT itself started as a centralised programme until a change in government transformed it into localised systems (Fishenden 2010).

Among these ongoing debates new approaches are stirring which are neither centralised, nor decentralised. Contemporary to social movements that try to influence politics, these novel approaches seem to confront the status quo and position themselves as legitimate alternatives. In some sense, their processes are similar to Coiera's (2009) and Robertson's et al. (2010) middle-out way, where neither the bottom nor the top are as relevant. In these processes, the local becomes communal, the needs are seen as transversal and are shared, and rigid organisational boundaries are bypassed. Exemplary of this trend of alternative approaches is the NHS Hackday (Todd 2014), where a group of doctors and computer hackers hack away at what they believe is wrong with the IT in the NHS.

Whether these are disruptive movements or fads is a matter that still needs to be settled. However, their existence points to certain trends of new collectives vying to become legitimate actors and contest established ones. openEHR is close to that world of alternative or middle-way solutions. It brings two main innovations: it is an open source project in a domain dominated by closed source systems and it deploys a particularly flexible architecture to define clinical concepts. This chapter presents openEHR as the case study of this thesis, and the wider context in which it evolves.

The sections are structured as follows:

EHRs in health IT. This section presents the history and evolving debates on electronic health records (EHRs).

The openEHR case study. Introduces openEHR.

The openEHR Way. openEHR's attempt to change EHRs.

5.1 Electronic Health Records and Health IT

Electronic health records (EHRs) are exemplars of the particularities found in the information systems health domain. The digitalisation of health services usually involves EHRs as the central part of the system. For example, the multi-billion NPfIT programme offered a direct route to patient registers and electronic prescription services among other patient-centered systems (Coiera 2007; Gillies and Patel 2009). EHRs, and their socio-technical implications, therefore, have a capacity to enable an evolution of IS use in the health sector (Takian and Cornford 2012).

Health records are a cornerstone in medical practice (Ingram and Arikan 2013). A patient's history, as told by patients, can be of relative quality and may need to be interpreted (Feero et al. 2008). Asking for and investigating this history is part of the doctor's training, forming the first step to the treatment of a disease. This information has not always been electronically stored but has existed for a long time in the form of paper. Hence, health records have usually been decentralised systems, each medical center containing their own, storing physical papers in file cabinets (Halamka et al. 2007). Interoperability between medical centers was (and still remains in many cases) the patient's responsibility, moving the necessary copies of medical papers from place to place. Historically, they were objects that moved little and stayed the same for long.

Electronic health records (EHRs) promise a wide range of benefits and efficiencies. The advantages are many:

- interoperability and standardisation (Bakker 2007; Burton et al. 2004; Häyrinen et al. 2010; Humphreys 2010)
- increase of patient flow (Clegg and Shepherd 2007)
- improved quality and patient safety (Ford et al. 2006; Hassol 2004; Johnson et al. 2007; Sittig and Singh 2012)
- efficiency and decision support (Daniel, Ewen, et al. 2010; Feero et al. 2008)
- patient-centric systems (Feero et al. 2008)
- reduction of costs (Stewart et al. 2010)
- fighting fraud (Simborg 2011)
- global research (Friedman and Parrish 2010; Kho et al. 2012)

Quickly surveying the benefits of EHRs might give the impression that they are a silver bullet for the health domain. The interactions of EHRs with realities is much wider than their paper counterparts, and the materialisations they promise different. New worlds are opened because of their digital substance: EHRs, for example, will offer chronic disease patients certain unique benefits to understanding and treating their illness. (Burton et al. 2004; Kho et al. 2012).

However, numerous concerns have been raised about EHRs. Despite being designed as patient-centric systems (Black et al. 2011), there is little that the patients know about their own EHRs, even where it is stored and who has access to it (Agarwal and Angst 2009; Bakker 2007; Kluge 2004). The implications of problems with EHRs are also unclear (Coiera et al. 2012; Grogan 2006). Some authors question how their successful adoption should be measured (Vest and Jasperson 2010). Others highlight that EHRs are not likely to completely drive off non-IT health records (Saleem et al. 2009), suggesting possible ensuing complications and questioning their completeness and consistency attributes. Even the promise of interoperability is questioned when contextual factors are brought in (Torre-Diez et al. 2013).

Besides, the implications of developing and implementing EHRs are not yet well-known. Only after implementation do questions about their management arise (Häyrinen et al. 2010; Prados-Suárez et al. 2012). EHRs are not abstract entities, but take different shapes and flavours depending on the context, the values and the implementation levels (Robertson, Cresswell, et al. 2010; Takian and Cornford 2012). They affect processes (Rajeev et al. 2010; Sittig and Singh 2012; Unertl et al. 2012), sometimes in different ways in the same organisations (Lanham et al. 2012).

Amid this polymorphism and malleability when trying to understand EHRs, it is not surprising to find a single, satisfying definition, despite there being a standard one (Hayrinen et al. 2008). At its most basic, an EHR could be considered as a digital record of a patient's health history (Chan et al. 2011). Such a simplistic definition hides the complexity of EHRs behind the simplicity of static files. When put in action, EHR definitions take on a myriad of shapes: patient-empowerment (Hayrinen et al. 2008; Mandl et al. 2001), questions about confidentiality and security in medical sectors (Anderson 2008; Salomon et al. 2010) or physician loss of control (Winkelman 2005).

That a standard—as EHRs could be seen to be—cannot accommodate most intended uses or decrease uncertainty as to the meaning and purpose of an object reveals its drifting nature (Ciborra 2002b). This is not only a matter of multiple perspectives, but that the realities in which EHRs are evolving are multiple. EHRs are digital artefacts that entangle themselves deeply into organisational settings, processes and roles, creating moments of instability. In some sense, they reveal the already existing rapid changes that occur in health, and how information systems must cope with them. As such, EHRs are not static but highly derivative objects: they enact realities through their materialisations.

The plasticity of EHRs is partly due to the malleability of the information and the knowledge that can be derived from them. This is a major difficulty concerning EHRs: they have semantic meanings. Even if they were stored as hard data, the need to interpret them certainly changes their meaning. How was a patient's tension measured? Seated? Lying down? Should the physician consider the standard method of recording tension only? In what context was the tension taken? Were particularly stressful events causing such numbers? Were these measures of insulin taken in mg/l, or in mmol/l (Mol 2008)? Because medical knowledge needs interpretation (Mol 2002), EHRs themselves inherit semantic meaning, whether accurate or not. They are not silent objects, but can be quite loud. Embedded in larger organisational settings, this multiplicity of meaning—and thus, of enacting realities—can be difficult to manage. To deal with it, openEHR takes a different approach to understanding EHRs: one based on multiple layers of information to which different actors give input.

5.2 The openEHR Case Study

openEHR is an open EHR system founded in 2000. Its purpose is not only to define EHRs, but also how these are created. It follows certain modelling paradigms that are, in part, a result of the founders' previous work. Despite its success (e.g. openEHR has informed the basis of several international standards on EHRs), the paradigms embodied by the openEHR Foundation are still controversial and are competing among many other standards.[1]

openEHR has gone through several major discussions while it was being researched and is still undergoing them. This research focused on the debates that occurred between 2009 and 2010. It is during these two years that new foundations for openEHR were put in motion. These new foundations involved long email

[1]http://www.openehr.org/about/origins

threads in the mailing lists concerning changes in the direction to better adapt to the evolving environments and internationalisation of openEHR. Chief among these discussions were conflicting views about open source and the search for financial partners. Open source is a recurrent topic in openEHR. It is telling for an open source project to negotiate what open source is. This is not uncommon: other projects have wondered about the purity of open source (Cornford et al. 2010). In openEHR, though, the concern was larger: open source was often discussed in terms of requirements. Part of this concern goes back to the project's early days when its idea of creating special knowledge requirements—archetypes—was first articulated.

5.2.1 Early History

The ideas behind openEHR came together through various initiatives that marked the experiences of many founders. In 1988, as part of the European Union's Framework Program on Advanced Informatics in Medicine (AIM), the Good European Health Record (GEHR) project was launched. The project was organised as a consortium of academic and industry partners across Europe under the leadership of Prof. David Ingram who would become a founder and the president of the openEHR Foundation (Kalra 1994). The GEHR won a bid and work began in earnest in 1992. The purpose of the project was to understand and review the clinical requirements that made a 'good' EHR. Defining EHRs in a flexible way would preserve the individuality of both patient and clinician, and avoid the over-simplification of treating them as an accumulation of dispersed data (Kalra et al. 2006). Formalising what a good EHR was made it possible to understand how to grow and 'care' for EHRs, almost in a phenomenological sense (Ciborra 2002b). In a way, this made them part of a process of construction and maintenance, giving them life as actors. From the discussions and the contentions, EHRs enacted controversies and created realities.

The GEHR project came to an end in 1994, but its essence stayed on and informed new projects in Europe which would largely influence the Commité

Européen de Normalisation (Chen 2009).[2] In Australia, some GEHR participants and future members which achieved some important leeway (the work from GEHR and openEHR have been fundamental to the Australian National E-Health Transition Authority). This period coincides with pre-standard endeavours and with the realisation that, for EHRs to move beyond the academic field, they needed to be confronted with implementation constraints.

The idea to create the openEHR Foundation rose from the concern for a growing divergence between working groups from Australia and Europe. The foundations' role would be to harmonise the different tendencies within the same paradigm. The Foundation would be open to all but conditioned to signing a set of principles. The official aims of openEHR are to:[3]

- "be open to all who sign up to its objectives and methods of work
- have free individual membership
- charge membership fees for official bodies, on a not-for-profit basis item help to define and support a common process of specification of clinical requirements, specification and implementation of systems and evaluation of the electronic healthcare records provided
- publish the work of projects and systems conducted within the openEHR community and adopting the GEHR methodology.
- offer the sources of such GEHR-based systems, in which IPR will be assigned to openEHR, under an open-source license within the community. Individuals or companies assigning IPR to the Foundation may where necessary and appropriate be remunerated under contract or through license fees.
- offer all its work openly in a spirit of a public enterprise, believing that this is the best and perhaps only way that appropriate high quality and interoperable systems are likely to emerge, worldwide.
- seek constructive relationships with groups and communities focusing on other aspects of clinical information management such as messages, terminology, knowledge-management and decision-support."

[2]CEN stands for the European Standardisation Committee in French
[3]http://www.openehr.org/about/origins

openEHR has taken a particular way in trying to standardise EHRs. Centered on open source, it provides not only the product (requirements, specifications, validators and tools), but also acts as a caretaker for the things that its community builds. In order to achieve this, openEHR attempts to develop a platform around which EHR-based health systems are founded. In this sense, openEHR tries to define EHRs in a clinical and technical way (modelling clinical knowledge in digital objects), but also from other angles: the sociological (common processes of changing specifications), the engineering (open source collaboration), the institutional (evaluation of EHRs), and the economic (search for partnerships). The mangling of all these concerns together gives rise to a pre-transition governance model.

5.2.2 Pre-Transition Governance

Despite being a young project, openEHR draws its roots from a long history of past projects and members' experience. It is different to already studied open source projects since it has had time to formalise its processes and its objectives through various iterations. openEHR, in this sense, is the culmination of certain beliefs shared by core members of the Foundation which are translated in its governance model.

openEHR uses a formal governance model, with a Foundation board sitting on top, composed of core people, most of whom were original members of openEHR and worked closely even before that on the GEHR or related, subsequent projects. Decision making is then divided into the Architecture Review Board (ARB), and the Clinical Review Board (CRB). Both boards aim at supervising and controlling changes made to the specifications (IT requirements) and to the Archetypes (business requirements) respectively. The reason for this formal structure is to ensure the proper satisfaction of the interoperability objective, and to avoid possible breaks to the system due to subsequent changes.

These two boards therefore are key to determining the strategic decisions of openEHR and which requirements, whether IT or business, are chosen. For that reason, Change Requests (CR) are channelled through these boards. There is, however, the exceptional case of unproblematic CRs which do not need to pass through formal acceptance by the boards, but will still be reviewed post-hoc.

5.2.3 Transitional Period

The transitional period is grounded in recurrent discussions that slowly gained momentum during the period in which this study is set (2009 – 2010). It is a response to questions that arose from the experience openEHR gained through the years, the realisation of the necessary diversity of actors to be involved, and openEHR's diffusion into new contexts. The questions driving this change revolve around openEHR's purpose, how it fits within the world, its relation to open source or its strategic partners, how to maintain financial stability and independence, and how to specify requirements. In essence, the transition marks a passage of maturity, from a Foundation that is managed and curated by core people, most of whom are original members, to one embracing new actors. The result culminates in 2012 when a new governance and IP licensing models are implemented. As part III will show, requirements were never far away from these discussions. The next section presents openEHR's conception of requirements.

5.3 Requirements and Processes in openEHR

The principal technical contribution to EHRs made by openEHR is its multi-level architecture. Such an architecture (similar to the model-view-controller in software engineering), allows clinical objects to be defined and constrained at various levels of abstraction. This means, in practice, that changes in some parts of the system should not cause changes in other parts, allowing for greater interoperability and distributability of requirements. In other words, the level of dependence between clinical concepts is minimised. In addition, by logically separating entities (thus creating ontologies), it is possible to attribute semantically meaningful knowledge to clinical models that can later be interpreted by actors (human or otherwise). This section briefly shows how clinical requirements are modelled in openEHR.

In openEHR, clinical requirements are labelled as 'archetypes'. These are versioned: they represent in sequential versions the latest clinical knowledge and the various ways to measure it. The immediate consequence is that requirements are understood to have a certain element of change. Instead of constricting that change, openEHR has built an architecture allowing changes in the way clinical requirements are expressed. This architecture, inherited from openEHR's ancestor projects, is termed as 'two-tiered' (Leslie and Heard 2006). This way of modelling requirements is seen as openEHR's principal technical contribution. The two-tiers refer to two main levels of abstraction: a Reference Model (RM) and above it an Archetype Model (AM); both serve as a basis for the construction of any particular clinical requirements.

RM models define the base classes for generic health records (Atalag et al. 2012, p.3). This abstracted level is generally stable and the foundation to which the second level, the AM, provides semantic meaning (López-Nores et al. 2012; Maldonado et al. 2012). The RM "consists of a small set of object oriented classes which depict the generic characteristics of health records (e.g. data structures and types) and the means to define context information to meet ethical, medico-legal and provenance requirements" (Atalag et al. 2012). What this means is that the RM defines the building blocks that will then be assembled and constrained to create semantically meaningful concepts. The RM tries to model the clinical world through the use of four conceptual classes which relate to actions in the world: OBSERVATION, EVALUATION, INSTRUCTION, ACTION (Garde et al. 2007). For example, and following closely the object oriented languages and their modelling, a foetal heart rate *IS AN* OBSERVATION (c.f. figure 5.1).[4] The *IS AN* defines the relationship between the abstracted conceptualisation of an occurrence and its type.

The second tier extends and constrains the classes and types that exist in the reference model, creating archetypes (Maldonado et al. 2012). Archetypes "aim to record clinical statements by specifying a set of domain-level concepts in

[4] `http://www.openehr.org/wiki/display/spec/openEHR+1.0.2+UML+resources`

the form of nested hierarchies of data fragments" (Allones et al. 2013, p.394). Using another example, the blood pressure archetype is composed of systolic, diastolic, and other measures, along with protocols (measuring at the thigh or arm, for example), states (whether sitting or standing), events, and descriptions. These measurements are semantically meaningful because they refer specifically to what is required to understand a 'blood pressure'. In the reference model, blood pressure is of type OBSERVATION, but adds clinically meaningful data structures that describe the observation. Archetypes then, enforce certain attributes and specify data structures. In this way, the archetype is interpretable by humans, and processable by computers. An archetype, therefore, is a static definition using the RM and AM. The definition is described either from the purpose built Archetype Definition Language (ADL), or an XML file validated through a schema. Archetypes, in a way, are themselves data units. They represent the "maximal data set for that given single clinical concept"[5].

To be contextually usable, they need further constraints (e.g. to be usable in practice, they must be instantiated by clinicians to their particular contexts (Dias et al. 2011)). These take the form of templates, and usually aggregate several archetypes to satisfy clinical use cases. For example, forms can be created that satisfy the Brazilian health service (Dias et al. 2011). As another example, the EHR of a person with diabetes would include parts of the 'Blood Pressure', 'Blood Sugar' and 'Drug Medication' archetypes (Allones et al. 2013), and may also form part of a nursing observation template (Wollersheim et al. 2009). Figure 5.2 shows the role archetypes take in describing meaningful clinical requirements that will be contextualised in order to satisfy specific clinician needs.

For openEHR and their members, archetypes are the principal clinical requirements (which will be subsequently used in specific contexts). They should represent the latest understanding of clinical concepts in a form that can be interpreted by humans and processed by computers. In that sense, they do not have an immediate functional character: they are a specification waiting to be used. To be imple-

[5]`http://www.openehr.org/wiki/pages/viewpage.action?pageId=1376257`

mented, an archetype needs to be further specified depending on the context of use and probably needs to be constrained through templates. Yet, the templates are dependent on the choices made in articulating the archetypes. This characteristic of archetypes is problematic and makes them the most interesting requirement to study, more so than templates or specifications. openEHR archetypes are developed through an uncertain and changing process of sourcing and articulating needs and understandings; the construction of a future need without having perfect knowledge of what that need might be. Archetypes' position as representatives of clinical requirements makes them the core requirement concept in openEHR. As clinical requirements, they share the same difficulties that any other software requirements suffer from, as discussed in chapter 2.

The versioned archetypes are stored in the Clinical Knowledge Manager (CKM). The CKM acts as a central repository for the community to develop, discuss, translate and publish archetypes. The discussions found in the CKM are specific to one particular requirement, and thus, relate only to that requirement. As discussed in section 4.2, the mailing lists, in contrast, question the role of archetypes and requirements, and the processes that develop them. The discussion goes beyond the definition of a single requirement and encompasses the nature of and the value of requirements to openEHR. If the study were to focus on the requirements in the CKM, then it would find that their nature goes largely unexamined. The mailing lists, in contrast, show how difficult it is for any actor to fully understand requirements and the influence they have beyond the representation of a specific need that awaits implementation.

5.4 Conclusion

This chapter has introduced the case study. The first section presented the context of health IT and EHRs in general, and emphasised the technical and institutional difficulties actors face. The following sections focused on openEHR. openEHR is innovative in the field, being open source and proposing a modelling architecture that destabilises established notions. openEHR's principal contribution is the definition of archetypes—clinical requirements which describe the semantic meaning of a clinical concept. These archetypes can then be interpreted by clinicians and processed by computers and represent the principal focus of openEHR's requirements work. Their value, the roles they take, and their articulation are informative to what openEHR understands to be the nature of requirements. The explicit attention the openEHR community pays to how they approach clinical requirements and their nature offers a suitable case for studying the broader requirements' processes. In particular, openEHR is concerned with requirements' processes that can capture, specify and share their potential.

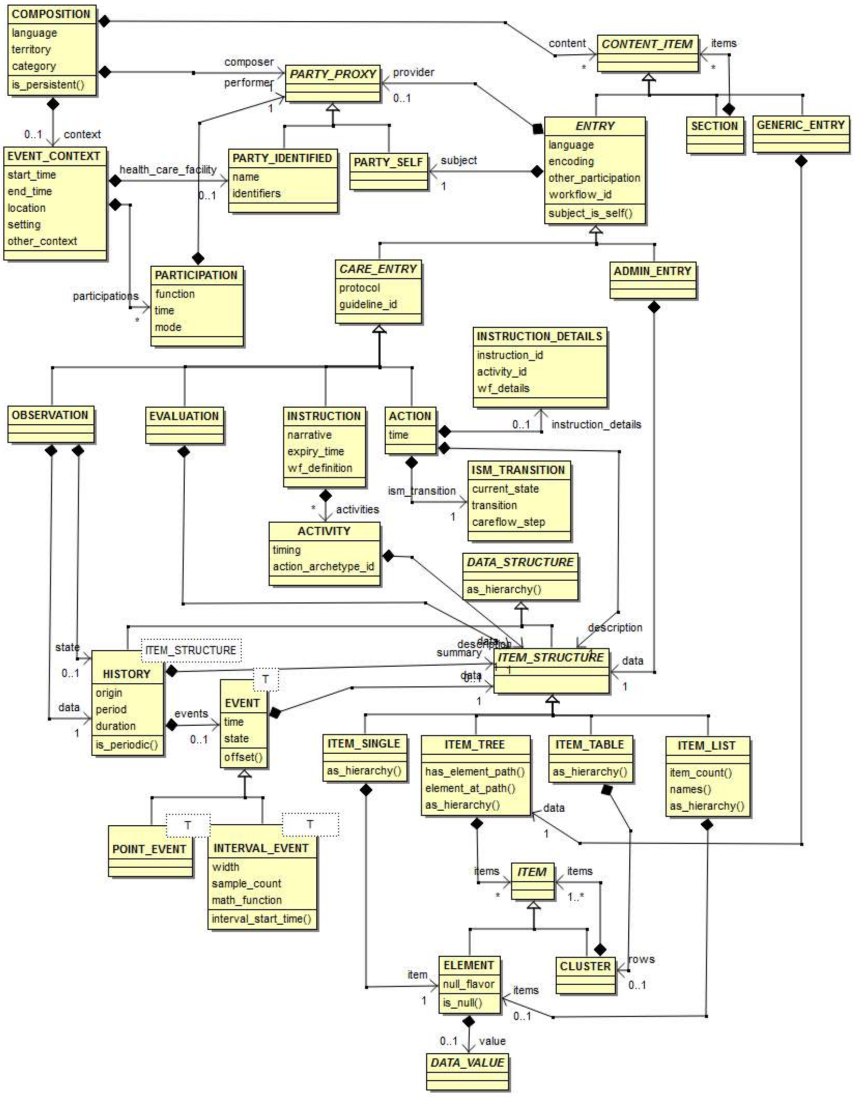

Figure 5.1: The openEHR Reference Model (from `http://openehr.org/wiki/display/spec/openEHR+1.0.2+UML+resources`)

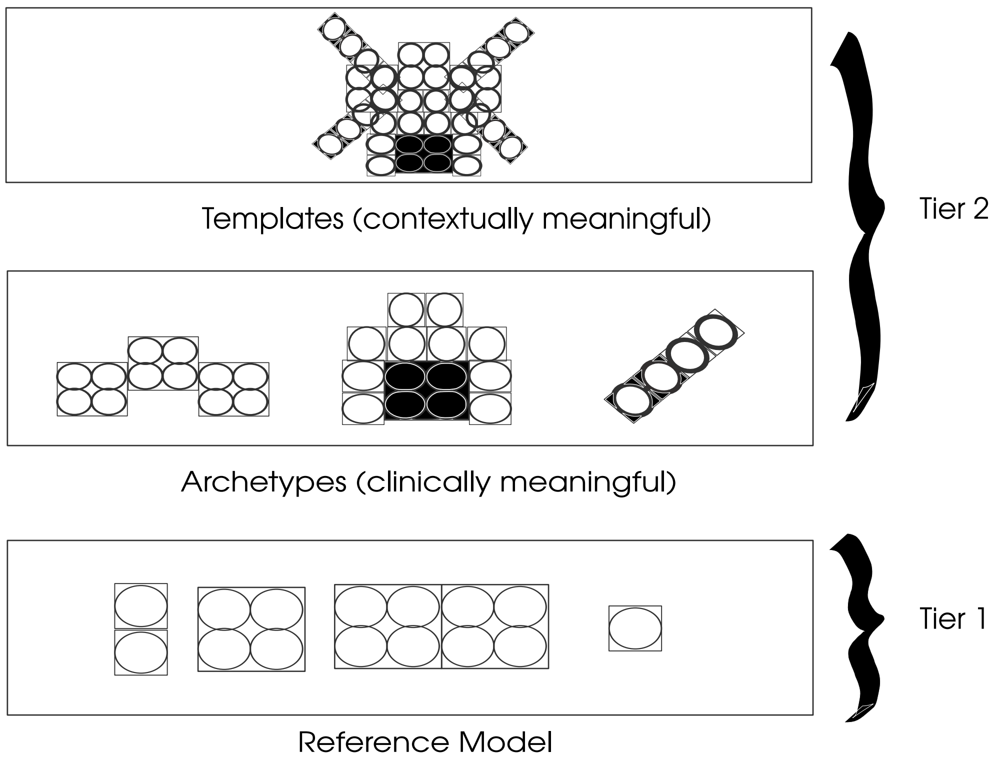

Figure 5.2: Conceptual view of requirements in openEHR's two-tiers (based on (Atalag et al. 2012))

Part III

Starting from the Middle

Mapping the Analysis

The findings of the study are analysed in the following three chapters. This short section functions as an introduction to the analysis, which is based primarily on the mailing lists. This data is then balanced with material from the interviews held with various openEHR members.

The analysis of the technical, clinical, and announcement lists shows the latent instability of requirements and the processes that enact and sustain them. This is evident in the controversies (of various strengths) presented, and which have challenged the project and its requirements and processes. This instability is due in part to the becoming (and thus multiplicity) of meanings and implied actions which requirements engage—an instability that is an essential property of becoming. This becoming causes an ongoing permeability of the project and concretely influences actors: the worlds explored; what the project should be; how requirements should be developed; who should be involved in creating requirements; what requirements are and the forms they might legitimately take.

To initiate the analysis, the findings synthesised from the mailing lists and the interviews with key stakeholders by being grouped into three principal movements and themes (c.f. section 4.3). The controversies recounted here do not generally have a specific beginning or end, although sometimes they take on a particular force leading to specific acts or agreements, such as the growing recognition of the need to change the governance model and requirements processes to better account for the growing community's needs. In fact, most issues addressed here have been skulking around the project since its inception.

That many issues remain latent or semi-resolved and may be re-opened at any time is, perhaps, an interesting particularity of open source software processes and their narrative practices. The three principal themes are: openEHR's *identity* (its place in the world of health informatics, EHRs, and its community); *territories and contexts* (the places where it seeks inputs and where it hopes to have consequence); and *modes of engagement* (where openEHR reflects how to pace requirements, what kind of process it should have to articulate them, and their meaning to the project). It is clear that these themes are intrinsically inter-linked, and often difficult to analyse apart. For example, a discussion on the license type of openEHR's intellectual property involves a simultaneous debate:

- over various modes and pace of engagements (stabilising core requirements or allowing greater emergent change)
- over the coherence of openEHR's position with respect to open source
- over the influence to openEHR's ability to diffuse its particular views of what an EHR should be

This entanglement of issues and debates, actors and interests, is revealing of the reach that requirements can have. Almost every controversy holds an entanglement of some form, making any straightforward rendering difficult.

In the chapters that follow, however, the primary analysis of issues is done within a single theme. In other words, even though a controversy could be used to express issues across several themes, they are considered under a particular theme and their links are highlighted when appropriate. For example, openEHR's open source identity can be used to explore certain contexts and territories in particular by diversifying the ownership of its intellectual property. As some participants argue, this can mean the emancipation of openEHR in contexts it could not have otherwise imagined it could reach. The larger, ensemble view is returned to in the final analysis chapter (chapter 8) on modes of engagement and in the discussion chapter that follows (chapter 9). Thus, each main theme has a chapter of its own:

Chapter 6 shows how, far from serving the the single purpose of sketching or freeing one reality, as requirements are typically understood, openEHR works to expose and accommodate the variety of existence(s) that requirements' need to embody (through a heterogeneous community). This role in holding on to diversity (as a potential) defies the single reality as in traditional requirements engineering. I thus argue that requirements are not boxed and dominated as the Cartesian tradition of control and stability would believe, but that they *are* in the world (Zizek 2012). Requirements are alive: they transform, influence, evolve, seem fragile one instant and strong another. They possess qualities of substance and expression which move in time and space and themselves shape substances and assemblages. This analysis adds a messy aspect to requirements: the project's identity has an influence on their sourcing and articulation. Requirements are, thus, not far from project politics.

Chapter 7 argues about the project's framing of requirements. Contrary to traditional requirements engineering which frames the project early on along generally clear boundaries, open source has no imposed time-line. Part of openEHR's requirements processes activities is to understand territory in terms of time. openEHR's requirements have to be shapeable in such a way as to provide a timely access to the articulation of clinical requirements. Yet, being an open source project, some participants complain about the lack of knowledge on the 'place' openEHR was, asking how could openEHR and its requirements be diffused to new territories? This has some direct implications to the contextualisation and control of requirements.

Chapter 8 reflects on openEHR's more direct actions on requirements processes, requirements, and the qualities and affects they try to give them. The two principal modes of engagement, emergence and stability, though seemingly at odds, are necessary at one time or another to sustain the requirements, their processes and their capabilities for evolution. Too much emergence can lead to destructive opportunism, too much stability can lead to destruct emergence altogether. Requirements, in this sense, cannot be said to be definite constructs, but instead hold potential. Any requirement has different substantive and expressive forces, some stronger than others (as is the case of the interoperability requirement); a requirement's assemblage will likely be different to another's. It is because of this that requirements can be in-becoming, enlisting their own assemblages, giving them multiple, potential meanings. Thus, because requirements can be multiplicities, there is no such thing as a 'pure' requirement. This chapter links the two previous ones and leads the way to the discussion chapter by proposing the notions of competing requirements multiplicities that hold different potentials.

Chapter 6

Identity: Being in the World

openEHR's presence in the world is not a given, nor a stable thing. openEHR's identity moves and fluctuates, sometimes unexpectedly so, with unintended consequences to requirements. That requirements can affect (and be affected by) a project's identity shows how intimate their relation can be to core aspects of a project (Doganova and Eyquem-Renault 2009). This chapter focuses on requirements as they relate to openEHR's efforts for *being* in the world. Using the verb *to be* when using Deleuzian theory might seem contradictory. But for Deleuze, *to be*, necessarily involves a function, and therefore, a movement. As Bateson argued, the acrobat balanced and apparently still on the high wire struggles to remain stable and moves to remain still (Farjoun 2010), and so does openEHR.

For openEHR, being in the world involves building and sustaining an identity. This identity is not built behind closed doors; it is related to the world at large and the various values, desires and needs the actors bring into the project. openEHR's identity continues to be developed as a matter of controversy, and this is necessarily associated with requirements since this identity is bound up with what openEHR can and should do. Associated to requirements, openEHR's identity is about what openEHR can and should do. It is a promise as much as a realisation of the moving ground on which openEHR stands. From its inception, openEHR had to argue for its own existence, though openEHR, as evidenced by the data, wants to do more than simply exist. openEHR's concern for *being* in the world is not easy:

the project starts from a minoritarian, 'academic' background which has come against established players. openEHR has to navigate these political currents, while remaining independent in its way of sourcing and articulating requirements.

The chapter is structured as follows:

Building a collective. Analyses the role and different notions of open source in negotiating openEHR's identity and the sourcing and articulation of requirements.

Back to basics. The influence of requirements on openEHR's origins and identity.

The minoritarian openEHR. This section analyses how politics influence openEHR's identity and requirements.

6.1 Building a Collective

This section looks at the issues that openEHR has faced in building its community, and how these issues influenced requirements. Open source projects share a particular concern for building a community. The success of open source projects often depends on its community. Indeed, the community is often responsible for the sourcing of requirements (Raymond 2001). Though often, requirements need to be asserted to core developers who, if interested, will volunteer to implement them (Iivari 2009a). For openEHR, the community has an even greater importance: it is not only a source of requirements, but a source for the articulation of requirements.

The discussions taking place in openEHR's mailing lists put into question the existence of a single, unitary concept of community. Even the term 'community' and the associated flatness of member relations is not clear (which is why this section is called 'Building a Collective'). There is, however, a prevailing interest by all participants to remain united (at least in objectives) despite, or because, of the heterogeneous ambitions and views about achieving them. Requirements play a center role in shaping these different views, which in turn gives life to various competing notions of community and how it should be governed. These notions revolve around the alternative ways to best source and articulate requirements. In this sense, the governing of a community is linked with requirements and the processes that articulate them. The competing notions of community indicate different understandings a community plays in the sourcing and articulation of requirements. This is related to the openness of the community, and how decisions are made as to what is a 'good' requirement.

Open source development is often centered around a core group of people and their lieutenants who act as filters for requirements (Cornford et al. 2010). In this respect, the 'open' of open source is more ambivalent than what it appears at first. Any one can take the project's resources—requirements, codes, documentation—and start anew, taking a different direction. Doing so, however, could result in a fragmentation of the community, to the detriment of all. There is, thus, an inherent motivation for the community to stay together, despite, and because of its heterogeneity and the advantages this can bring. In question then, is the project's 'open source' governance, and its influence on the sourcing and articulation of requirements.

To work together, some participants felt the necessity to bring the boards' focus on their needs, and how they could achieve them:

> "Governance is a good issue to discuss with the community, but I can't see any governance if the OpenEHR boards are distant from the community, and do not understand their real needs."
>
> Pablo Pazos, 'Why is OpenEHR adoption so slow?', technical list, November 2010

At other times, it is Ocean's close involvement with the openEHR foundation which is regarded with suspicion:

> "So but the open source is the open base but openEHR you have open source, open source deployment and not open source, because they are Ocean, it's not open source but it's open based. So it's a difference. And that just was always a problem in openEHR that a lot of trouble that you had because just many of the tools that are in openEHR were done by Ocean Informatics and use Microsoft technology. Then there is the Swedish one, the American they did

> this Java implementation and openEHR is more and more and more open."
>
> openEHR adopter

Often, open source is associated with freedom from vendor lock-in. Indeed, because the source materials can be reused and taken in different directions, open source is, in general, not dependent on any project, foundation or organisation. Controlling the articulation of requirements through the use of tools, whether Ocean's or Microsoft's, leads some to worry about how free and open requirements really are:

> "Yeah, they think: "hold on a second, if I'm going to use openEHR I'm going to have to use Ocean Informatics tools, that's not completely open.""
>
> openEHR clinical lead

Although, not everyone is of this mind, which goes to show how heterogeneous open source communities can really be:

> "I don't understand how the terms "vendor lock-in", "behind doors" etc. are associated with Ocean - I see it as a social service organisation rather than a commercial entity by the way :P
>
> Interesting times ;) Though I am hoping that we can identify the underlying reasons why some of us got so much sensitized and discuss this openly..."
>
> Koray Atalag, 'openEHR community on Google Wave', technical list, December 2009

Still, the relations between actors in the community does not count for everything. There is a varied and willing community behind openEHR, "happily" engaged in communal work:

> "Still people are happily engaged in the work, there is some kind of community trust, which is a nice thing. Some companies with close connections to the foundation also seem to be comfortable with using these archetypes within their products and services, nice

> for them. I believe this proves that there might be an interested community even under very unclear licensing conditions and that they don't seem to mind if their contributions may be used commercially without a licence guarantee demanding derivative works also to be open."
>
> Erik Sundvall, 'License and copyright of archetypes', technical list, October 2009

The quotes from the openEHR adopter and the clinical lead indicate a problematic reality for requirements: the resulting ambivalent identity about what is 'open' obstructs the adoption of openEHR, which in turn obstructs the building of a collective and the further sourcing and articulation of requirements. On the other hand, there is a sense of community. Despite the diverging views, the mailing lists are plastered with calls to review and comment on requirements. In play, is a negotiation between different views of what open source is, how it relates to openEHR, and what kind of community should ultimately participate in the articulation of requirements. In this sense, an open source community is not a given, it is built and negotiated through compromise.

For some, open source is a natural means of engaging the community to achieve rigorous requirements:

> "Remember I've coined this phrase of rigour, engagement and trust. And the rigour bit, for me, in the scientific world, most of physics couldn't exist without open source software, because that's the way people know, you know, software is extraordinarily complex, unless you've actually got it in your hand and you work with it [...]"
>
> Foundation Board member

So, open source is not only about building a community, but of building the right kind of community which will be able to engage and articulate the right kind of requirements. Open source, in a way, is itself a selection process for requirements. This is curious because open source, perhaps naively, is usually portrayed as an inclusive, communal engagement where all things go. openEHR shows this to be true, but only in certain aspects: by discussing what is open source, how should openEHR's community be like, the participants in the project are affecting what openEHR is. The project's requirements, and their becoming are carefully and explicitly curated for quality:

> "In both examples, a relatively small number of people do decide (in a technical development environment) what is a correct solution to the problem at hand. In the case of Linux, Linus Torvalds is famous for being autocratic - but it works. This is life, not everyone is an architect. The number of designers at BMW is but a tiny fraction of the overall payroll. If it were any other way, we would have chaos. These efforts then offer their output to the world at large, and the world at large decides. Both IETF [*Internet Engineering Task Force*] and the LAMP [*Linux-Apache-MySQL-Perl/PHP/Python*] platform are massive successes.
>
> That is because they did not decide on what was /correct for us/ - we did that - we decided what /worked for us/."
>
> Thomas Beale, 'ISO 21090 data types too complex?', technical list, November 2010

There is a correct becoming in openEHR that pulls the project into following the original, established philosophy. There is a core here—which indicates there is also a periphery—that needs to be protected. Just like other technical projects, open source or otherwise; there is a need for a technical core to establish some contours and allow for useful participation. For some, the explicit aim for the community to engage with the project with "rigour", as a Foundation board member said.

But the core and the periphery are not necessarily divided along the lines of the technical and the clinical. Quite the contrary, the core can also reflect a general, but distinct, "self-selecting" group of *doers*:

> "There's things in life, what you find then is that people who talk and people who do are not always the same people, you're very very lucky if you've got the talkers who do a lot, and unfortunately, good doers don't often talk a lot. Hmm, sorry I don't want to be over general about it, but there is a practical reality that things make progress that people doing things and they're a self selecting group really, and then a lot of people are worried, OK, this is not democratic, or whatever, or this is not, you know.
>
> I'm not against democracy, but there's an issue about what is the driving force and where you're trying to get to and you have to

> be really quite supportive of the people who are doing things and you have to protect them from some of the, you know, other side of, the way that gets viewed in certain quarters. So I think, hmm, I think we've sort of done the best we could along the way, given that we didn't get a lot of high level support for it."
>
> Foundation Board member

What the board member argues here is that core people are those who *do*, and they need some level of institutional protection. The process governing the self-selection of people is a certain kind of meritocracy, though less elitist perhaps than those found in other open source projects.

All the work being publicly available, open to all eyeballs[1] looking for bugs and inconsistencies can be a burdensome load to carry. Without some form of recognisable blessing over the work being done, the core might become fragile, unable to set in motion the core requirements it sees as important. This is an open source "practical reality", as the foundation board member says. Protecting the core people means also protecting the sourcing of requirements, as well as protecting the correct procedure for articulating them. In many ways, protecting the core is a hands-off work since "they're a self selecting group". The governing of the requirements process does itself through the *doers*. Almost like a species of their own, they recognise one another.

openEHR's *doers* are "the driving force and where you're trying to get to and you have to be really quite supportive of the people who are doing things". In many ways, it seems, openEHR depends on this core to become what it desires. This views the process of sourcing and articulating requirements through the support given to those who *do*. It is not a heavy-handed control, it is based on a meritocratic support system which legitimises certain requirements over others. Requirements are not pushed forward, but guided through the support of the community (or an institutional aspect of it at least). In this respect, harnessing innovation in openEHR means internalising requirements by providing institutional support to the self-selected group.

[1] Raymond's so-called Linus's law: the more eyeballs, the less bugs (Raymond 2001)

It might seem that the "self-selection" of groups and requirements means there is no need to tender the garden, that the *dooer-cracy* grows 'naturally'. The community, though, has a protecting role of its own:

> "Won't your feared modified redistribution only be a problem to interoperability if, all the following comes true:
>
> [...]
> c) If national programmes/authorities etc. will start telling people to use the "commercial" versions instead of the openEHR ones for national exchange use. (Or more likely they would start their own repository for international archetypes [...]
> d) If the really valuable clinical community creating and maintaining archetypes etc. stop supporting the work in the openEHR repository in favour of other alternatives.
> I think c and d would only happen if openEHR messes up their governance and/or community support, and if that is the case, then it is actually a good thing that the community, using CC-BY, can take the archetype collection and keep innovating elsewhere. CC-BY might actually pressure the openEHR foundation to do a better job than if feeling too "safe" behind CC-BY-SA [...]
> The more formal power you try to cling on to, the more informal power you risk to lose."
>
> Erik Sundvall, 'License and copyright of archetypes', technical list, October 2009

The core and the periphery of the community is again redefined. Sundvall's argument is also about protecting a "self-selecting" group of *dooers*, one which is also core. They too are "creating and maintaining archetypes". However, they do not necessarily have a formal role in openEHR's institutions. The role of that community is to be vigilant to the direction openEHR takes, to what requirements they are sourcing and articulating, and what role the non-institutional community takes in that process of sourcing and articulation.

If an open source solution is not good enough, the community is able to sound a forceful alarm without much effort: it just has to stop using it. Such force can act as a "brutal" control mechanism by the peripheral (non-institutional) community that Sundvall proposes (Biddle 2013). In some way, Sundvall asks for a renegotiation of the role that peripheral *dooers* take: by giving more potential force to the peripheral community; Sundvall institutionalises the peripheries. In so doing, the periphery can come to own the requirements' articulative force.

The community is a diverse group which also includes companies and their users. For others, it is not the *dooers* whose requirements need to be protected, nor the periphery's, but the users'. As Ocean's president, Heard worries about the risk that another company might sabotage openEHR's efforts to create interoperable EHRs. What if another company could copy and copyright openEHR's requirements if protected by less restrictive forms of copyright. Protecting requirements, for Heard, means protecting his clients, his company, and the openEHR foundation as a whole:

> "So the tension here is between companies using archetypes being able to secure their investment in the software they produce and no one feeling threatened to use archetypes in their system. [. . .]
>
> The point is the collective investment in archetypes will be massive. How do we deal with the situation where someone creates a good archetype as a base idea and posts it on openEHR. Then someone specialises it quickly on the web and copyrights the archetype saying this is their archetype and no one else can make one like that? As someone said earlier on the list - these are all our collective ideas and it is inappropriate for anyone to claim them. But we have to have a collective governance structure that works and supports the processes that support communication of health records."
>
> Sam Heard, 'License and copyright of archetypes', technical list, October 2009

The risk, according to Heard, is the possible imprisonment of requirements and the detrimental effect it could have on the users. His point resonates with Stallman's defence of Free software as opposed to open source.[2] However, this view also sees the building of an open source community as needing some kind of protection for its requirements. The difference is about which aspect of require-

[2]The main difference is Free software's licensing obligation which forces any derivative work to be licensed under the same, original Free license. Open source does not restrict the sharing of its code and requirements, it allows anyone to make derivative works without the obligation to share back.

ments should be protected. Sundvall is less concerned about the physical protection of the requirements, he does not see the need for a stronger protective license. What Sundvall is more worried about is the community's force to express its own requirements without depending on central powers. Heard is more concerned for the physical, substantial (as in material substance) well-being of requirements. openEHR's requirements, he argues, should remain openEHR's. These are two different views on requirements which have a direct influence on openEHR's ability to build an open source community.

Open source mediates for requirements and forms part of requirements' assemblages. Through this mediation and the creation of a community, open source helps define the landscape in which requirements should be sourced and articulated. Open source is an ambivalent concept and openEHR has multiple ways to build a community. There is no single, unitary concept of what an open source community is, and how it would affect requirements' becoming.

6.2 Back to Basics

Requirements discussion, at times, addresses the project's foundation and original purpose, to reassess it and place it in contemporary dimensions. When a particular thread questions the modelling of demographic archetypes, it shows the tight, entangled relation between requirements and the project's core. This tight relation raises some questions: are requirements owned by openEHR? Could openEHR be said to be its requirements (e.g. constituted by them)? These questions show that there is, to some degree, a mutual, *in-becoming* constitution between requirements and openEHR's identity. In other words, openEHR and requirements apply forces on each other. However, they are not the only actors. Requirements and identity often find themselves entangled with other assemblages such as history, process or open source in a *manglish* way (Pickering and Guzik 2009).

The revealing controversy arises unexpectedly after one of the list members browsed the demographic archetypes and found the editor software giving out continuous errors. The issue was raised quietly with a simple question on the technical mailing list and asked whether the problem was with the archetypes or the editor. Out of this question stumbles a variety of considerations that give rise to a heated debate. It is surprising to see a controversy about specific requirements—the status of demographics archetypes in openEHR—reveal itself as such a fundamental issue for the community. The controversy finally comes to an evolving resolution: to some, the core of openEHR remains unchanged. To others, a new recontextualisation was becoming part of the core and called for new requirements.

The controversy starts innocently enough when a participant reporting an error from the archetype editor writes:

> "When I try to open it with the AE [*Archetype Editor*], I do get continuously error messages similar to: [*image as attachment showing an error message from the archetype editor*] Is there anything wrong with the archetypes, or is this an error in the archetype editor.
> Anyone els experiencing such problems?"
>
> William TF Goossen, 'Why is the editor not opening ADL files?', technical list, March 2009

The problem, as another participant responds, is that the demographics archetypes had been hard-coded and the editor does not support demographics archetypes any more. This is a requirements question: are demographics supported by the tools to be used to create other archetypes? In other words: is a particular type of archetype supported by the tooling? Nonetheless, not long after the start of the thread, Goossen takes up an earlier question and changes the discussion topic to which types of requirements should be considered legitimate.

> "> How many more types of archetypes are we envisioning to support?
> >
> I think the tools need to support ANY archetype that represents valid content in health care."
>
> William TF Goossen, 'Why is the editor not opening ADL files?', technical list, March 2009

There is an explanation for this limitation, however, and Heard, an openEHR founder and Ocean Informatics president, explains the historical reasons. The reference model had been hard-coded into the editor at the time before implementing the flexible ADL. In any case, the demographics requirements were not in the scope of an EHR to express.

> "The Ocean archetype editor was built before ADL and works by using the ADL parser and XML parser. The reference model is hard coded as a class. TO work with another reference model it is necessary at the moment to hard code another class. A generic approach to this is possible and may in time be useable. In the meantime, I just want to be clear that the demographic model archetypes cannot be used in the EHR - they are not relevant there."
>
> Sam Heard, 'Why is the editor not opening ADL files?', technical list, March 2009

A quick reply from Verhees follows and momentarily reroutes the discussion from an apparent attack on Ocean, to being grateful for Ocean's work. The community, it seems, can quickly reassemble itself into real world, organisational collectives.

> "It is also clear, in my opinion that Ocean has no obligation at all to build any tool. It cost time and money to do so. And the community must be grateful for what it gets. Me, I am. There work of ocean (and many others) give me a wonderful business opportunity."
>
> Bert Verhees, 'Why is the editor not opening ADL files?', technical list, March 2009

However, this inability to define demographics requirements in the EHR still worried some, and the discussion continued, away from ownership and power concerns.

> "I am confused - hopefully you saying that those particular 'older' demographic models are not supported?
>
> But there are newer ones right being added to the CKM that conform to the ADL structure the other clinical ADLs use?
>
> You are not saying an AQL query for
>
> Women over 50 - 70, last mammogram > 2 years

cannot be supported because demographics are not relevant?"

Greg Caultron, 'Why is the editor not opening ADL files?', technical list, March 2009

Others took this up going so far as questioning the purpose of the project and recentering requirements as depending on the overall goals, and prioritising them according to specific contexts:

> "Maybe I should ask a question first. Are we building a model JUST for personal healthcare; or for general healthcare? I ask this because I get the impression from Thomas' statement (and the overall all direction of archetypes. That the mind set is that healthcare information is ONLY personal.
>
> [...]
>
> While as an American I love the ability built into the openEHR model to separate demographics from clinical; in the real (larger) world, in many cases, that demographic information is CRUCIAL to healthcare systems to see what is happening in a region."

Tim Cook, 'Why is the editor not opening ADL files?', technical list, March 2009

The previous emails expose the assemblages that lurk behind specific requirements. They associate demographics requirement and their existence to the governance of openEHR. In question is the relevance of certain requirements in specific contexts and the overall direction and application of openEHR. The discussion does not target the specific demographic requirement (although it is discussed); it is an overall examination and recontextualisation of openEHR's relevance and coherence. In other words, the worry is that openEHR might not be targeting the contexts that some participants thought it would and that openEHR is forgetting an important reality.

> "this below - demographics not relevant in the EHR is like the most confusing comment ever I heard from you.
>
> About whom are we going to create a EHR then? If it is not possible to have the individuals name, id, birthdate and sex in the EHR (generally named patient demographics), it becomes useless in my vue.

> Or do I miss a point here? "
>
> William Goossens, 'Why is the editor not opening ADL files?', technical list, March 2009

What takes place in this particular thread is a reconsideration of what the basics are. To do this, Goossen cites stable, accepted standards on the definition of EHRs:

> "Hi all, it is sometimes good to deal with the basics again. ISO 18308 defines the EHR as a virtual collection of patient related data. It assumes the data are stored for an identifyable individual in more than one technical system (virtual).
>
> ISO 18308 also assumes that the data can be used accross the continuum of healthcare, including continuity of care, management information, epidemiology etc. "
>
> William Goossen, 'Why is the editor not opening ADL files?', technical list, March 2009

Another participant, Beale, reveals another context to discover: the application of legislation to technology criteria, which has (perhaps) forced the separation of the demographic model from the core of the EHR RM. The demographic requirement has always been present, but because of legislation, has not been the integral part of the EHR.

> "yes, in fact, what he meant was that the more comprehensive demographic information modelled by the openEHR demographic model is not directly in the EHR, it is in dempgraphic information objects, which would typically be in their own service - particularly in Europe where legislation largely requires this to be the case. This does not of course mean that demographic information of clinical significance would not be in the EHR, e.g. date of birth, sex, occupation etc."
>
> Tom Beale, 'Why is the editor not opening ADL files?', technical list, March 2009

Heard supports Beale's argument and states that, in practical reality, it is advantageous to separate demographic data from the core EHR.

"We are taking a service oriented approach here and it has benefits. You can store demographics in the EHR if you like and there are archetypes to allow for this. But generally this data is stored elsewhere in real systems. It is used for administration, billing and a lot of other things. Many hospital have large systems before they implement the EHR. So the EHR as a service can exist without recording the demographic information in the EHR itself. In France it is illegal to do so ? requiring separate security to get access to demographic and personal health information.

[...]

I hope this helps. It might not seem intuitive but it is very practicle and a result of a lot of companies being involved in the architecture design (all with different demographic services and requirements)."

Sam Heard, 'Why is the editor not opening ADL files?', technical list, March 2009

There is a strong practical reality and sense of hard implementation conditions in Heard's email, which dictate the 'space' where the demographics requirements should be modelled. It is experience of the world that instructs putting demographics away from the core of EHRs because it is so desired by implementers. Experience legitimises the space demographic requirements take and why the core is how it is.

The community in open source, however, is an heterogeneous assemblage and other participants have their own experience to share. Community is not a united, simple entity. Even unitary concepts of core and users, although present, give way to more complicated assemblages. What is core, as the previous messages have shown, is not as stable as it may seem at first glance. The same goes for the concept of community and its identity. openEHR as a whole is put in question by the demographic requirement. Demographic requirements start establishing new boundaries along unexpected lines, through space and time. The project's identity, and its purpose, are on the move.

"Sam, I often brought this subject up [demographics in openEHR], maybe five times last year, the answer differed from, 'We'll add demographic archetypes to the ArchetypeEditor within a year' to now carefully stating in a direction that demographic archetypes

> will loose the relevance. I know Sam, the latter is your position for longer time. I believe, we have to distinguish the position of Ocean as a company, and the definition of openEHR as a worldwide concept/standard.
>
> As I said before, I don't think that Ocean should add demographic archetypes to their tooling. It would be nice if they do so, but that is their choice. They are a company, and have their own priorities.
>
> You are addressing WIlliam [Goossen] in this message, but it is a message on a public mailinglist. I see your message as indirectly addressed to the community. So I am not answering for William, I am answering on my own behalf."
>
> Bert Verhees, 'Why is the editor not opening ADL files?', technical list, March 2009

The demographic requirement is restless. It invites itself into discussions in which it might not make technical sense to be, but that would in a clinical and political context. It shifts the boundaries of what is technically the core of the project to a larger political and clinical meaning. McNicoll touches on this, when proposing various ways to accommodate demographics in different spaces. The core of the project, thus, is permeable to a larger assemblage than 'pure' technical criteria. The core identity leaks a bit, and this leaking invites a negotiation of the other actors in the assemblage. The enunciation of the demographic requirements forces the articulation of wide and deep sets of assemblages, touching the root of what an EHR is or should be, and of openEHR itself.

> "In principle, I absolutely agree, but if in an existing PAS [*Patient Administration System*] is to be interfaced/mirrored, we must accept that the faciliites e.g to record carers, may simply not be available. As an EHR vendor, I can see the attraction of just getting on with modelling this in my EHR.
>
> >TB [*As written by Thomas Beale in a previous email*]
> >but this is faulty modelling - archetype is not the place to do such modelling - it is being misused in this instance to express the
> >requirement / aspiration of having a form on the screen with a fully populated demographic items. The requirement is reasonable of
> >course, but needs to be modelled elsewhere - it is the job of a virtual EHR service to integrate information from EHR, demographics and
> >other services and present it on the screen in appropriate forms.

> Again, I cannot disagree but whilst the openEHR architecture and tooling has been largely successful in appopriately obscuring the technical aspects of modelling from clinical modellers, this is an aspect that we have not got quite right as yet.
>
> First off we need to be able to develop Demographics model archetypes via clinically useable tools (which is happening and is where this thread started) and then we must be able to display them within the tools in an understandable and contextually appropriate fashion.
>
> Whilst much of this could be done at template level, there are situations where, from a user perspective, this is counter-intuitive. As I said before, the strength of openEHR has been to be able to present clinical information in an immediately understandable fashion, no matter that the underlying technical model is quite different."
>
> Ian McNicoll, 'Why is the editor not opening ADL files?', technical list, April 2009

Requirements are not still things, they touch on individual's values, incite reactions, influence actors' affects, propose new ways to be shaped. A requirement is a force of expression which articulates a substance, such as the EHR. Values are one of the things requirements use for doing this. Combined with the open source nature of the project, are freely discussed what seemed (to some participants) fundamental and established aspects of openEHR. In open source, it appears that a project's relevance and coherence can be discussed at any time.

6.3 The Minoritarian openEHR

This section starts building a bridge between the project's identity and the territories visited by openEHR. Such a bridge places openEHR in its most fragile yet proudest state: as a 'minoritarian' project trying to achieve its goals of interoperability in health systems. The section narrates an account of openEHR as an academic project trying to make the jump to become a recognisable alternative to established EHR projects in the field (the territories). The incidence of this jump on requirements is important: the type of project that is built and the ways it is represented and showcased will affect the kinds of requirements it attracts. Requirements seem like a window through which to trust or distrust the project. Much of the concepts openEHR put forward shifted the grounds on which important players had founded their work and forced openEHR to justify itself and its motives repeatedly. This section analyses the incidence of openEHR's minoritarian identity and general politics on the sourcing and articulation of requirements.

openEHR starts to walk on its own in 2000, but still remains attached to its parentage. When discussing openEHR's foundational requirements, it is difficult to avoid talking about its parent GEHR. The GEHR had also been, in its time, the subject of counter reactions. Its requirements, and the qualities it thought to deliver to the field of EHRs were seen as disruptive and dangerous:

> "When the GEHR project came to an end, a reaction set in against it. The first GEHR Object Model became a focus of concerted opposition within some groups working in the domain in the UK and within CEN [*European Committee for Standardisation*] standards bodies. The questioning of the assumptions and approaches of existing products and approaches had been inevitable, but did not make GEHR popular, although its approach always sought to be constructive. It was unfortunate that the work of GEHR came, apparently, to be perceived as a threat to other interests and ambitions, evidenced by the manner in which it was opposed, early, tentative and incomplete as its results were."
>
> David Ingram, openEHR Foundation president, 2002[3]

Much of openEHR's recent labour has been to transcend that categorisation as a dangerous outsider and to be instead represented as a legitimate player proposing a constructive—if alternative—solution. openEHR's careful construction of requirements and their subsequent liberation (available openly) was a statement to its ambition of becoming a philosophical paradigm *for* interoperable EHRs, and not aimed against anyone:

> "Contextual (i.e. use-case specific) features should always be added in specific classes / locations in models dealing with those specific use cases.
>
> The openEHR data types are designed like that - it is just basic god practice. They can be (and are) used in messaging, storage,

[3] `http://www.openehr.org/about/origins`

> GUI, business logic. Context specific features are modelled and coded where they are relevant, not integrated into what would otherwise be completely generic data types.
>
> Not understanding this basic modelling practice has lead HL7 to produce models that are very far from being easily implementable or reusable - which is a real pity."
>
> Thomas Beale, 'More on ISO 21090 complexity', technical list, November 2010

As an outsider, entering a highly political and institutional domain, openEHR has struggled to make itself heard. It pushed itself on various fronts to prove its relevance. It used its academic setting for financial support and legitimacy:

> "I think that the roots of openEHR are very academic and very big picture. You know, the idea of having a language that you can use to help develop health systems is a very good one. So the archetype development language is excellent. It needs... academically it's spotless because it's been very well thought out, it's been studied, it's been validated academically, so it would be excellent, an excellent basis to start any project with."
>
> An openEHR satellite project member, interview

The academic legitimacy that made openEHR's requirements so attractive had an unintended effect: it gave the appearance that openEHR was a detached project, distant to the immediate necessities on the ground. A disconnect was perceived that could only be solved through important changes: by shifting the attention from one space to another. Such a relocation would allow the consideration of important territorial realities that the academic space could not be sensitive to, and thus discovering 'habitable' worlds unknown to openEHR.

> "[. . .] but the majority of openEHR work that is out there is in academic space. It's not in the front-line space. That must massively change."
>
> A clinical board member, interview

The focus on the 'front-line' could be reminiscent of the top-down or bottom-up discussions in software engineering of the 90's. But how to be at the 'front-line' and still maintain generic requirements? To the board, it is through: “implementation, implementation, implementation!”

> “The participants in the recent openEHR list debates have reflected widely different perspectives and interests, but a broadly shared goal. Caricaturing the dialogue a bit, it’s all about:
> standards, standards, standards;
> and/or open-source, open-source, open source;
> and/or governance, governance, governance.
> Whereas, as far as /open/EHR is concerned, it’s all about implementation, implementation, implementation!
> Of course, the underlying drivers cannot be wholly detached from considerations of power and influence, reputation, resource and personal freedom.”
>
> David Ingram as openEHR Foundation Board, announcement list, December 2010

There are several ways to balance between the ‘golden core’ of generic and contextualisable archetypes, and satisfying local concerns. Though bridging the high-level and the local can be problematic for openEHR’s requirements, there are several ways to consider the ‘front-line’: several formalising levels (international, national, local), deformalising control (capacity for local reaction without waiting on higher levels), competition between several openEHR-like archetype repositories, a gold repository that could be curated by the Foundation, etc. In all these options, requirements remain at the center of the problem: how to develop a business model that can best cater to openEHR’s and the community’s needs? The consequence is a project-wide questioning of what the project is. For example, should it be a curated library?

> “The analogy that comes to mind is the interaction between publishers and librarians. In the context of librarianships, you have national repositories [...] you have some kind of governance framework around the numbering and cataloguing [...] and you have an ecosystem of publishers. You need a new kind of governance which recognises the curation, the librarianship, the skills, is an analogy related to books, there’s going to be a correlate of that in the context of archetypes, templates, and there’s also going to be a world of

> publishers."
>
> openEHR board member (interview)

If various levels are considered, how should they be formalised?

> "We are discussing how we establish localisation chapters system. To be a global standard, localisation is necessary. However, bad localisation would disrupt all."
>
> openEHR technical member and implementer (interview)

There is a political aspect to requirements which can help or hinder a project's capacity to grow onto new territories. Identity is one such political aspect. openEHR's identity, and how it relates to the world affects its capacity to push, support or control certain requirements in certain domains. Each of these different processes for sourcing and articulating requirements holds different potential for their *becoming*. The different potentialities give openEHR several choices to become what it wants to be. Whatever 'it' that may be, it must articulate 'real' requirements anchored in space:

> "There are always different assumptions, different things done, and you can bolt them together, but it doesn't mean that you aren't going to get new different behaviours. So you actually have got to have the platform in the space and something that's fast moving as requirements are changing, you know, all of the tools of the trade are changing, if the software is all buried in the bake it holds you up in commercial and business ways, but it holds you back in scientific ways as well. So, that's why I think [*open source is*] important."
>
> openEHR board member, interview

Behind a marked aura of objectivity, of scientific modelling of clinical models, hides in plain sight *ways* to make the requirements process more relevant. The above interviewee recognises the political nature of requirements. If requirements were not considered in careful commercial and business ways, it would affect negatively their scientific nature. This is because (or a cause of) "different assumptions".

'What *is* openEHR?'; the question about its identity becomes blurred with: 'What can openEHR *do*?' Multiple *becomings* that, nevertheless, need to be materialised:

> "So for instance, I would like to have an international group who look after emergency medicine archetypes, but it's very hard to engage those guys at the moment, because what do they do? There is nothing to do, there is no archetype. Ok, you can agree on an archetype, but then, what do you do? You can't use it, so... It's not going to survive."
>
> A clinical board member, interview

The processes that substantiate requirements are important. For openEHR, much of it is based on a certain level of trust that its minoritarian identity allows, and even, demands. There is a 'right' way of defining clinical concepts, and open source is part of the necessary ingredients:

> "You can't have rigour if you're not able to exchange the tools and the methods in a practical and implementable form, so I, I think there's a sort of a reason, I mean physics just could not have done things in the Large Hadron Collider unless it's been, all the numerical methods, all the handling of data, all of the processing of data, all of those things that's been exchanged in open source. So, that's, that's why I think..."
>
> A clinical board member, interview

Although openEHR yearns to enter the mainstream political arena, its 'minoritarian' and its challenging attitude sometimes leads to confrontation:

> "Rather than working toward a common pathway to mutually promote both HL[7] and openEHR, we have spent a lot of energy of the negatives of HL7.
>
> If I became the one source of standards, I think I could make the perfect standard. Of course, no one else would think so. As openEHR expands it use, it will get (and has gotten) pusgback from persons who think it does do what they want it to do. Then openEHR can say tough luck or they can change to accommodate. Now you are in the world of HL7."
>
> Ed Hammond, 'HL7 modelling approach', technical list, November 2010

openEHR's 'minoritarian' attitude could be said to be 'anti-political'. Not that it is not political, but that it combats actively the political status quo, to the risk of annoying key actors in the field.

Always in a minoritarian *becoming*, it seems openEHR tries to avoid any force that could overpower its independence, even to the risk of being portrayed as anti-democratic. It guards closely the quality of its requirements and looks for processes that legitimate requirements. However, as openEHR becomes a bigger player and feels pressure to represent its interests in more contexts, openEHR has to expand. In so doing, it must consider processes that could negatively influence the recognised academic quality of its requirements.

6.4 Conclusion

In this section, an account has been made about the relation identity has with the sourcing and articulation of requirements. Instead of being only 'objective' objects, requirements prove to be full-fledged actors, aligning into complex assemblages and enacting a world on their own terms.

When openEHR builds a collective, it must negotiate between various understandings of what EHRs are and between the various identities living along in openEHR that push and pull the project in different ways. Sometimes, it is a requirement and its meaning that question the project's identity. When openEHR tries to represent itself to the world, it has to deal with its cultivated reputation as a 'minoritarian' and academic project, to challenger of established players.

The surprising reach of political aspects to the sourcing and articulation of requirements influences the forces that materialise them. The consequence of this alternative view of requirements is to question their material composition, and how the various forces play out in trying to shape their substance. Open source, for example, using requirements to mediate its own values, has made the re-evaluation of requirements less difficult, alighting them of some of the over-powering sources, 'liberating' them, as it were, into discussable actors more prone to emergent forces. Some of these new forces are based on the contexts and territories that openEHR explores to find and articulate requirements. Bounded, strangely, to both core and unexpected, peripheral requirements, openEHR tries to make sense of the contexts and territories it should travel to. This play between the core and the periphery, the original purpose of the project, and its attitude and relevance compared to its competitors influence the sources available for openEHR to articulate its requirements. The processes that need to be put in place are altered, and openEHR has to rely on its own initiative to counter the claims of more dominant players that they alone can understand the requirements of the field.

Chapter 7

Exploring Contexts and Territories

This chapter asks one question: how does openEHR contextualise its requirements from territories? Territories and contexts are used somewhat interchangeably since either term is hard to conceive without the other. A context needs a territory, and the territory needs to be contextualised and made sense of. The two words, if taken together, suggest a performative action on spaces: spaces need to be adopted through their contextualisation.

The previous chapter showed openEHR's origins in academia and its minoritarian position. How should openEHR move definitely beyond the academic bubble? This ambition implied a necessarily tentative approach to understand clinical contexts: 'who could be an ally?', 'who could be a competitor?' In other words, openEHR needs to come to grips with the 'real' clinical spaces; spaces in which it wants to make a difference and evolve. These spaces are often competing, creating different realities, and openEHR has to find competing ways to grasp them.

The settling of territories has a direct influence on how requirements should be sourced and articulated. openEHR looks into the strategies, tactics and issues that openEHR faces when measuring the 'real' spaces and how best to implicate itself in them. However, the territories and the contexts that live in them are not easily drawn. There is no single map of these territories waiting for openEHR, so it needs to develop its own cartography and use it to find requirements.

This chapter studies the maps that openEHR tries to draw. It is structured as follows:

Unseen boundaries. This section analyses the resulting difficulties of defining boundaries. To do this, it analyses the influence time has on obfuscating boundaries. In that sense, the 'spaces' openEHR is exploring are not static; they cannot be drawn and used as references. This section looks into three different aspects of this problem: non-existence of boundaries; how openEHR analyses the spaces despite the lack of boundaries; and openEHR itself not being clear on its own place amid those open spaces.

Realising territories and contexts. This section reflects on openEHR's appreciation of its own position within the moving spaces it tries to settle and cater for.

7.1 Unseen Boundaries

openEHR's goal to step beyond the academic bubble and settle 'real' spaces proves difficult. The territories move rapidly, complicating any concerted efforts to identify and prioritise requirements. There is no single, unitary territory to which openEHR could move, no single reality to cater to. Instead, there are multiple places from which openEHR could define requirements and multiple ways to do so.

Time here is essential. Inside the academic bubble, openEHR could spend any measure of time with little consequence, but in the 'real' spaces, it needs to adapt quickly to changing territories and contexts. This changing environment is problematic because there is no clear, uncontroversial way of defining where requirements start and where they stop. This section looks at three aspects that, under the auspice of time, are problematic for openEHR to understand the territories and contexts it wants to be part of. First, that the territory and contexts are themselves moving. Second, that in order to define requirements in such a rapidly changing environment, there needs to be a process which takes time into account. Third, that openEHR's own boundaries—where the project ends and the external territory begins—is not clear.

7.1.1 Timing the Territory

A territory must be enacted (delineated, guarded, observed, allocated, etc.), or as Deleuze would put it, *territorialised* (Deleuze and Guattari 1987). Latour explains this 'effect' when describing one of his hiking expeditions, and, seeing a tower from afar, tries to guess its shape and size (Latour 2013, p.61). The tower takes different tones, appears squared at first, but ends up being round the further he walks. The space and its qualities change depending on the time spent mapping them. Time adds its own element to the 'unseen' quality of territories (not yet realised) by forcing actors and assemblages to be apprehended, *apprivoiser*, at different moments and velocities. Over time, the 'same' territory is unlikely to be seen as being the same twice (although, some elements will evolve less), and to this extent, makes mapping an atemporal activity. Mapping does not end, and the map, thus, is a process: 'measuring' the landscape, mapping, 'measuring' the landscape again, mapping, and endlessly onwards trying to perfect and sustain the connections, despite or because of their movement.[1] The mapping is never entirely satisfied, and neither is the measuring.

Because time would not allow it, there is no single depiction of a requirement which can last without modification. Even the most generalisable aspects of openEHR used to describe archetypes and clinical models are prone to evolve as they become entangled with other actors:

> "openEHR 'cooperating more' and not 'reinventing' would have been impossible without time travel. The openEHR data types were started in 2002, and were in production use in Australia in 2004. Since then nearly all changes have involved refactorings of functions and abstract types. If you doubt this, see the revision history of the spec"

[1] Measuring is used here in its largest definition possible: the relation of one reality to certain measures.

Thomas Beale, 'ISO 21090 data types too complex?', technical list, Nov 2010

Beale is worried about an external event, out of his control,—the legitimisation of ISO 21090 guidelines and its statements regarding data types—and the consequences for openEHR were it obliged to comply with them:

> "The guidelines *could potentially be disruptive to many projects that are underway*, either due to contractual requirements or expectations made to the key stakeholders.
>
> Additionally, there is a need to develop appropriate information mechanism (e.g. GForge site) to facilitate the integration of the ISO 21090 data types in a way that does not add any risk to the project and also does not compromise project's business needs.
>
> What I take from this document:
> * ISO 21090 is seen as complex, unvalidated, and risky
>
> I would be interested in hearing of other experiences."

Thomas Beale, 'ISO 21090 data types too complex?', technical list, Nov 2010

openEHR, even though promoting itself as an alternative, and therefore divergent (at least philosophically) from other actors and their ways, is connected and continuously moved by them. This does not mean that openEHR must change its position, but it is affected nonetheless. However wide the philosophical difference, there is no gap between openEHR's boundaries and those of other important territories.

As in this example, it is not time on its own which changes requirements, but the evolution in the assemblages of various actors (e.g. the ISO work). As Beale points out, some still hold the belief, despite the publication of the requirement specifications (the "specs") and the history of changes detailed in the issue tracker, that openEHR has not cooperated sufficiently, and thus has not committed itself sufficiently to this significant territory. The necessities of that territory are measured distinctly by different participants in openEHR.

The consequence of untimely requirements (requirements disconnecting out of time), is that they may disrupt existing projects. Some requirements might be more problematic than others, hitting openEHR hard, changing the territory and its requirement and breaking through what was once thought as established and well-defended boundaries. Boundaries appear static only to the extent that the project moves along or contrary to them. The implications of the ISO 21090 standard could be "disruptive" to openEHR's existing requirements and their processes.

Mapping the territories takes time. The actors employed to measure the territories and find requirements in them can influence how long, or how accurate those requirements can be. In other words, the territory does not conform to single realities. There are multiple ways to see and contextualise territories. Certain contexts will make it easier to achieve some work, while others will make it hard if not allied to certain actors:

> "From what I've seen, it seems to me that openEHR would work well on most levels of health requirements, however, if at least one of the big guys (IT, government, health orgs etc) doesn't get behind it because of their vested interests or inertia, you are going to be pushing it up hill. I think that a relationship with IHTSDO[2] could be an important one as long as you don't get subsumed into their agenda rather than pursuing your own."
>
> David Neilsen, 'ISO 21090 data types too complex?', technical list, Nov 2010

Time in openEHR is conceived in several dimensions. One is the time that it takes to become majoritarian (as Neilsen argues above, it would be easier to ally with the right actors for this). Another is the timelessness that openEHR wants to offer clinical requirements. So, the way the processes used to map territories influence what kind of requirements can be described. The processes may be at odds:

> "It needs to help me at the front-line, to solve a clinical, solve that problem now. Not in five years' time, or in ten years' time, it needs to solve a problem now. But, I'm willing to use it at the front-line, from a bottom up way, in line with the top down approach, if we can align them. But more than anything it needs to work bottom

[2] A non profit organisation which owns and maintains SNOMED CT, a terminology project which defines low-level, biological concepts.

> up. If you had to choose between bottom up and top down, I think it should be bottom up."
>
> Clinical lead, interview

In some way, the interviewer above presents a modest goal (though not achieved by any system yet): to align to current realities, far away from concerns of future-proofing and interoperability. openEHR should solve a problem right here, right now. Space and time put together. Time, here, forms an obligatory passage point demanding an immediate attention before anything else may be considered, and if it were down to choosing, the interviewee would choose timely rather than timelessness. After all, future-proofing, with all its extra complexity for requirements, will only solve a problem if it *actually* solves a problem. Time, here, disconnects and distorts territories, distorting in turn reality. To *actually* solve a problem opens the door to a multiplicity of territories. The territory is not to be blamed, it is *through* it, by imagining multiple actual realities that the multiplicity is created.

The creation of a multiplicity puts openEHR in a bit of a bind. openEHR's concern for future-proofing clinical contents allows it to satisfy requirements in a variety of similar, general contexts, but in none in particular. In other words, openEHR offers the potential to solve many problems (some which had been invisible precisely because it unites contexts together). How should openEHR measure its potential and act on the actual? One way is to locally measure territories *and* unite with other local territories, despite their possible divergence:

> "Let's say there are ten emergency departments in ten different countries, and they all want to use archetypes, are you going to say that they can't make changes until the international organisation say they can make changes? That's not going to work, so you're going to have to allow some peer to peer sharing of good quality archetypes."
>
> Clinical lead, interview

Time is stretched, it allows and forbids all the while being qualitatively different to the interviewee's front-line and international organisation. What to make of this alienable conception of time? Modern physics has shown that time is relative, and therefore, indivisible. It cannot be packaged neatly into wholes and parts. It is time which allows and restricts the potential of requirements. The existence of the disconnect between the actual and the potential requirements is only, according to the interviewee, an error of measurement by openEHR's core concerning the proper contextualisation of time in the front-line relative to the project's own. In other words, front-line openEHR is telling headquarters that they are only jet-lagged and should normalise their time along a more propitious meridian. What is the difference between the front-line and the foundation's conception of a project?: the actuality and potential of requirements. Requirements are the elementary notions of change: it is through them that the mapping is measured and the territories and contexts settled.

For a mapping to be useful, it must hold a shareable identity. In other words, the mapping has to mean something (even if different somethings) to actors. For the requirements as mapping, this is the basic notion of what a requirement is in a particular territory and context, and it is maintained by its identity. A requirements identity is a 'commonness' which allows both the front-line and the project to recognise resemblances between requirements (both in time, and in space). To be able to say: 'this is an openEHR requirement' means that, relative to the requirement's multiplicities, there is an identifying factor; a mapping, so to speak, that leads to finding requirements. Malabou's investigations on destructive ontologies prove instructive on the matter: "For Deleuze, a true metamorphosis would be a metamorphosis that, despite its name, would have nothing to do with a becoming-form. According to him, "as long as there is form, there is still reterritorialisation".[3] This is why "becoming-animal" is not "becoming *an* animal": the first is an arrangement, the second is a form, which can do nothing but freeze a becoming" (Malabou 2012, p.17).

The identifying element of a requirement is not a form, at least, not only a form. Certainly, it helps to assign it with a name, or a description of a function.

[3]The "reterritorialisation" reflects a possible loss potential by reinforcing the territory over others

Only, there is much more behind the curtains. What the curtains hide is both form and expressive force which together create that qualitative identifying element. This, added to the assemblages to which it is connected, is the reason why requirements can be transformed in time through contexts and territories, and yet remain recognisable. Requirements have a plastic quality to them that allows them to be recognised.

7.1.2 Matters of Time

According to openEHR participants, time splits the sourcing and articulation of requirements largely into two camps: one camp follows a design-by-committee process (HL7 and other SDOs); another camp follows, to some degree, an open source philosophy.[4] The two camps contend over a main point: the legitimacy that the process gives to the exploration of requirements territories.

The design-by-committee is a top-down, legitimate work, well recognised by the institutional players:

> "I would like to see a process in which we fully and completely define the requirements for the standards we need. We debate, discuss and compromise. A small group of technical expeerts create the standard and then everyone evaluates if the requirements are met. HL7 has established a huge presence in the world. It would seem to me to be foolish to ignore HL7 when creating a datatype standard. As long as you have your standard and I have my standard, we have no standard.
>
> I think it is important to examine our motivations - what drives us in our work with standards. Is it a life-time work, or is it simple a detail that must be accomplished before we can do what we really want to do. Is our work with standards our claim to fame. There are times when I think HL7 has so many groups because we want a tribe of chiefs. Even that is driven by real requirements - my boss won't pay for my participation unless I have a titled job.
>
> You claim that ISO is flawed. Ballot is by standard, a only a few countries dominate. That obviously is not restricted to standards. Again, that's life.But what is a better solution? Shall we live with a decision making prosess in which a relative few people decide what

[4]Some participants question the degree of open source, or even what it is and what it really means to openEHR. Some see in open source a normal consequence for a meritocratic process, other are closer to the sharing and openness principles.

> is correct? While I like that approach, if I am a decsion making, I don't like that approach if I'm not."
>
> Ed Hammond, 'ISO 21090 data types too complex?', technical list, November 2010

It is, in the eyes of HL7 members, the only, even if imperfect, practical way to "fully and completely define requirements for the standards we need". It is a democratic process where votes are cast into "ballots", despite the "tribe of chiefs" it ends creating. It is a linear approach, principally top-down where a "small group of technical experts create the standard". Once presented, there is a bottom-up *reaction* after their evaluation. According to this view, requirements—their sourcing and articulation—are only legitimate via the consensual agreement of a project's members (e.g. grouped in a committee). The committee is the principle source of their articulation. Once legitimised, the requirements are pushed out into the world.

Those responsible for their implementation (and who are not members of the project's committees), are often limited in their input and can only comment after they are legitimised, and thus, hardened. This is an essentially top-down process, not only in terms of decisions about who participates and how they participate, but in the general control and ownership of the requirements. Once hardened, a part of a requirement's flexibility is gone. Those at the bottom have little force over changing and evolving requirements. According to Eric Browne, this kind of process politicises requirements:

> "I certainly believe that the whole ISO process with respect to health informatics standards is deeply flawed. As Grahame implies with the datatypes standard, the process is politically driven and compromises in modelling, engineering, safety, implementability inevitably occur. The question is how significant are these compromises and what effect will they have on the evolution of e-health?"
>
> Eric Browne, 'ISO 21090 data types too complex?', technical list, November 2010

The consequence of the politics on requirements, for Beale is that design by committee process does not provide a "realistic" anchorage between requirements and reality. The politics deviates requirements away from reality:

> "I would say it is a reflection of the committee-driven standards building process. There is no realistic prospect of physicians or anyone else going back through gigabytes of medical data and trying to decide what might have been in the mind of the person originally recording the data, and how that might have been somehow different from the terminology's definition of the meaning, due to some knowledge about how they chose it. There may be reasons to record the value set / ref set id, and it may be true that the author of the data could only choose some higher level code e.g. 'parasite infection' rather than 'giardia infection'. But whatever term is chosen, it must be by definition the correct thing to record in the data, otherwise the software and clinical process is a nonsense."
>
> Thomas Beale, 'More on ISO 21090 complexity', technical list, November 2010

The solution for Beale is to specialise work. This is, according to him, the only way to create a meritocratic process that would source and articulate 'real' requirements. Open source is there to organise a division of labour. Hammond disagrees and believes in a more democratic process to capture consensual(-ised) requirements. Otherwise, by excluding people based on their skills, parts of the problem space might also be left aside:

> "Your experiences are different than mine. I must confess that I have not been in any HL7 meeting which included shouting down. Disagreements - yes; but not shouting down. In any case, The problem with kicking out the non-modelers is that I am not sure who defines the experts - we ourselves? I am not smart enough to think I know all the answers - or perhaps I should say I am smart enough to know that I do not know all the answers. I strongly believe in the democratic process. My knowledge of other approaches have always lead to disaster. mI also have no problem with disagreements. In fact, I think we find better solutions when we disagree."
>
> Ed Hammond, 'HL7 modelling approach', technical list, November 2010

Time matters here. It is the principal actor in the sourcing and articulation of requirements. It is time's relation to the space where the requirements need to act that can cause a disconnect between their potential use and their actual use.

For Beale, the problem is not the group of experts making decisions, it is how these experts were chosen in the first place. He wants a meritocratic process to avoid the politics, and to avoid the politics, there should not be pre-designed groups of people who will end up becoming the principal source of requirements and those who legitimise their articulation. In others words, the sourcing of requirements directly influences their articulation and open source's meritocratic organising allows *any* participant (with recognised skills) to contest the requirements. Otherwise, there is a risk of over-engineering needless realities:

> "I don't have a problem with the fact that use cases have been researched, indeed that is an excellent achievement. I have a problem with the way the standard has been created on that basis; it tries to do way to much, incorporating all kinds of special cases that should not be included in the core generic data types standard. Such use cases can be accommodated in one or more separate 'packages' that show how to use the core classes for the specific purpose, and what extra classes are needed to formalise other data specific to the use case."
>
> Thomas Beale, 'More on ISO 21090 complexity', technical list, November 2010

There is a 'locality' expressed by Beale, both in terms of time and space. There is a right place for certain requirements to be expressed, and also a time. The important thing for Beale is to have 'core' requirements that can later be extended depending on local necessities, but which still conform to that core. Requirements can only become 'real' when the right needs have revealed themselves since there is no point in trying to cater for them before. This becoming of requirements, its time and place, is dependent on the implementability of those requirements in becoming. This ultimately influences the legitimacy of requirements by forcing their articulation to a later time and a less uncertain place, closer to their implementation:

> "I would also concur with your statements about the ENTRY sub

> types, as Sam mentioned we have built an INSTRUCTION index that tracks the current state/care flow step of instructions and their associated ACTIONs providing efficient access to this information. The effort required to implement this would have been much greater if these classes were not specifically modelled. I guess as openEHR penetrates the market, the more likely CEN 13606 would be updated with these enhancements. To be honest I think this is the right standards process, standardising of implementations that are known to work in practice. I am sure we will learn more and improve the ENTRY subclasses further before they go into the CEN standard, then the standard will be more useful."
>
> Heath Frankel, 'openEHR-13606 harmonization CR regarding CLUSTER/TABLE etc and ENTRY/OBSERVATION' (Was: ISO 21090 data types too complex?), technical list, November 2010

According to openEHR participants, it is by physically joining requirements to the specific territory in which it needs to evolve (in time) that generic requirements should be contextualised. To specifically question the evolution of requirements in time and their place in reality means questioning the disconnect that took place in the traditional requirements engineering process, even the Agile ones. openEHR is saying: 'requirements do not become in a limited time-frame'. openEHR does this by playing with the generality and the local-contextual specificity of that requirement (its realisation in the territory). This is similar to object-oriented designs which establish interoperability from the parent down to the more specific objects by relating the two through inheritance (though openEHR takes this further by making that inheritance evolve to the specific implementation contexts).

In openEHR's approach, there is a relativity of time compared to the space the requirement is in. Time is disjointed (not disconnected). In other words, the early conceptions of a requirement is potentially far away to its specific realisation in actual contexts. This disjointness is an atemporality for requirements, leaving them time to become, escaping the traditional requirements engineering linearity or sequential iterations. Seeing requirements as evolving represents another way of sourcing requirements and managing uncertainty: only when there is a *visible* need for a requirement should it be engineered. This is only possible if the previous requirements follow certain guidelines so that new additions down the line are compatible.

The consequence of this atemporality is that the concept of expert changes. It is paradoxically stronger and weaker. It is weaker because the expert is not institutionalised, so other experts (based on skills) may contest his opinions, opinions which are not institutional any more. The notion of expert is also stronger: the domain of expertise can be aligned with the requirement in question. There are more experts, with more diverse backgrounds. Overall, it is the expert's role which has changed to accommodate the paradox: there is less projective power, less oracle work about what and how requirements should be.

7.1.3 A Frontierless State

Time moves and shifts territories and changes the way openEHR must approach them to articulate requirements. But what are the implications to the project itself? Given the absence of deterministic boundaries, how does openEHR define itself in terms of space. In other words, what are the consequences of the lack of clear external delineations? The mailing lists show evidence that starting in the middle, having too much open air can be destabilising. Perhaps too much freedom is claustrophobic? Because the territories are moving, some participants complained about not knowing the 'state' of openEHR. The most immediate consequence is a problematic level of knowledge about the project's wider context which affects the participants' capacity to act:

> " For example, here in Latin America, almost nobody knows about OpenEHR in the industry area, and very very few knows about it in the academy area.
>
> There are some ideas that may help de difusion and adoption of OpenEHR:
>
> - I think that regional OpenEHR communities are needed to empower the adoption and spreading of the standard. In 2009 I send a message to the mailing lists, but I get no answer from the community (this mail is below).
>
> - In order to help any goverment adoption of OpenEHR, the decission makers have some questions that today OpenEHR can't

answer.
- What is the state of the standard?
- Is it stable?
- Wich parts are stable?
- Is there any return of investment study done on efective use of OpenEHR?
- Or just, how much time and money I have to spend to effectively use OpenEHR in a real world application? (I have to train people to make things happen, not in an investigation project, but in a production project)
- What real world products are using OpenEHR?
- How these products are using OpenEHR?"

Pablo Pazos, 'Why is OpenEHR adoption so slow?', clinical list, November 2010

The email seems almost phenomenologist. Pazos' call is almost a plee for a Body (in the Deleuzian sense). It asks: 'What are openEHR's affects? What parts of openEHR are stable, and what do they do?' Requirements, here, are the potential affects; affordances between the project and a reality. The question Pazos asks is a becoming one. A multiple becoming between his own experience and the wider project's experience. This problem is *almost* phenomenologist were it not for the consideration of distance in the phenomenon's creation. Time lends a hand again, not by itself, but through the proxy of distance.

Talking of affects, Bergson writes: "Our perception, we said, indicates the possible action of our body on others. But our body, being extended, is capable of acting upon itself as well as upon other bodies. Into our perception, then, something of our body must enter. When we are dealing with external bodies, these are, by hypothesis, separated from ours by a space, greater or less, which measures the remoteness in time of their promise or of their menace: this is why our perception of these bodies indicates only possible actions."(Bergson 2010, p.126)

Deleuze would make sense of this problem through the transformative potential of 'Bodies Without Organs' that are in part extended from Bergson. Distance is crucial to the body's capability to enact the world. From the physical spaces that time allows to be created depend the ease with which requirements can be contextualised by openEHR's participants, enacted in the territories. In other words, openEHR's actions, structure and processes have an internal incidence on its further potential to act. Time allows a certain distance to be negotiated, pulling and pushing contexts and territories closer and further away from the project. For an affect to develop, a body has to be present:

> "My opinion is the grade of adoption of a standard depend in some aspects of goverment agencies, in some of the industry and some of the academy.
>
> DICOM is a good example of an open standard heavily supported by the industry, that's the point of it success. Can't be OpenEHR a de-facto standard for EHRs? Like DICOM is for imaging. I think yes, but the progress of OpenEHR to solve real the problems and make it usable, is slow.
>
> I think OpenEHR is strong on the academy area. It has poor industry penetration (I mean enterprises developing tools and aplying a good part of the OpenEHR specification in their systems, and that these systems where used in some hospitals). I don't know what's the penetration of OpenEHR on goverment agencies. There are some open tools but there is some stillness on making improvements on them."
>
> Pablo Pazos, 'Why is OpenEHR adoption so slow?', technical list, November 2010

Requirements peer behind Pazos' words when he says: "the progress of openEHR to solve real the problems [*sic*] and make it usable, is slow". The lack of swiftness from openEHR to solve problems and become rapidly usable affects its market penetration and its position as a potential "de-facto standard". In question is the need to resolve a governance model which can better respond to local needs of guidance while maintaining a strong central link. The process of sourcing and articulating requirements makes openEHR "strong on [*sic*] the academy area", but less so in "industry" since there is no 'body' there to express anything.

The discussion is large, but centers on an essentialist question: what is openEHR? Beyond an identity, openEHR is also a place (and other potential places) that the project should adopt.[5] If the place is the center, satellite issues can gravitate around

[5] Adopt in the sense of *apprivoiser*, to create ties.

it and bring that territory closer, for example, through education, communication and tooling. In other words, the discussion stretches to those requirements not immediate to openEHR's hand, but that need to be pulled closer to openEHR so that it can achieve its full potential. For that potential to be realised, other realities must be taken into account: the financial difficulty of creating open source software.

> "Clinicians and developers need tooling to take control of complex concepts, and not having enough tooling is leading to lots and lots of angels and pinheads type of discussions. The chain of problems go like this: not enough tooling -> not enough implementation -> not enough understanding & feedback -> lots and lots of hypothetical discussions.
>
> So if (at least according to me) the biggest problem is tooling, why not build the tools and solve the problem? Because no one is paying for it [...] with limited resources it is hard to trigger a mass adoption."
>
> Seref Arikan, 'Why is OpenEHR adoption so slow?', technical list, November 2010

Arikan places requirements in the middle of an assemblage here. There is a need to understand the space through more "implementation". From there, other realities can be grasped, internalised by openEHR. Without anchorage in a territory, there would be "lots of hypothetical questions". The internal state of openEHR is put into question to allow it launch and articulate the territories and internalise its requirements. The territories are not stabilised, neither openEHR's own place nor what lies outside (except in hypotheses).

This section argues for the intricate relation between time and the contextualisation of territories. Requirements have no permanent boundaries from which they can be gathered as if they were a flock of lost sheep that needed its shepherd's voice to become part of the project. Requirements, on the contrary are so entangled, that it is difficult to think them astray. More than requirements, it might well be their shepherd who is gathering himself among the evolving fields. Time plays a role in the shepherd's hardship.

7.2 Realising Territories and Contexts

The previous sections have shown that territories and contexts move in time. This section looks at how openEHR evaluates the territories and contexts in-becoming, and how this becoming alters its concept of ownership. That boundaries move, that openEHR is not a unified singularity reveals relations between territories, their contextualisation, and the articulation of competing realities. These competing realities center around a requirement's 'value'; its acceptance into the project is dependent on the territories that become contextualised. A requirement 'foreign' to the project might not be so well received as one from an accepted source.

Concretely, the analysis focuses on how openEHR values offshore territories, and adopts its language to describe and articulate requirements. Once again, the distinction between sourcing and articulation thins.

7.2.1 Contexts *in-becoming*

The distinction between territories and contexts is approximative because territories are apprehended contexts. It is the uncertain nature of territories that necessitate their apprehension. Were the territories fully certain, then the project would not have to make sense of them. Apprehending territories is about coming to terms with the territories' particularities and contextualising them to the specifics of the project; to internalise them. Traditional requirements engineering often views this internalising through an analysis of the 'as-is' (the state of the current world), which is generally done at the beginning of the project.

The 'as-is' is a strange amalgamation between concerns of reproduction and decontextualisation and recontextualisation. The 'as-is' reproduces the world, reconstructing an artificial island where the project's requirements will cover some unmet need. This unmet need is an exercise in evaluation: it asks what necessity is not resolved in the world at that precise moment, therefore requiring an exploration of the territories. In doing so, the 'as-is' analysis both ties territories down, and deviates from them in an assumed and controlled way, to imagine a context where, at some point in time, it will become useful.

In openEHR, there is no such 'as-is'. The as-is is not evident because many of its requirements are 'plastic'—they have generative potential beyond one specific context of use and do not force their implementation. The unmet needs that openEHR has been working to cover have been in gestation for years, even prior to openEHR as GEHR's nascent incarnation. Yet, the territories in which openEHR has tried to settle into have changed; both openEHR itself and the territories to settle have evolved. Because openEHR's requirements are so 'plastic', it is forced to evaluate repeatedly the nature of the territory and the work it does to map the territory out. Among the debates concerning open access policies in academic journals, an openEHR board member makes a contemporary analogy:

> "The analogy that comes to mind is the interaction between publishers and librarians. In the context of librarianships, you have national repositories [...] you have some kind of governance framework around the numbering and cataloguing [...] and you have an ecosystem of publishers. You need a new kind of governance which recognises the curation, the librarianship, the skills, is an analogy related to books, there's going to be a correlate of that in the context of archetypes, templates, and there's also going to be a world of publishers."
>
> An openEHR board member in a board meeting

The cause of this perpetual re-evaluation is openEHR's plasticity, derived partly by its requirements substance (not forcing a specific implementation). Its place in the world has not been set in stone some years before. Instead, it has been grasping at ways to contextualise the world in accordance to its own set of beliefs. It seems strange to relate beliefs, territories and contextualisations, but the absence of an objective truth forces openEHR into evaluating its position in competing realities.[6]

The competing realities are a primordial and evolving mapping of openEHR's positioning in the world which makes it all worthwhile. How should openEHR solve this problem? What is this problem? *When* is this problem? *Where* is this problem?

[6]Or more precisely, the production of subjectivisations of truth produces competing realities (Guattari 1995).

Without this evolving mapping of territories and contexts, openEHR would only be able to give a single answer at a discreet instance of time. Instead, openEHR is *about* building an open source EHR. This *problematising* by openEHR opens the door to question openEHR's efforts to contextualise the world. To openEHR, 'reality' is an evasive multiplicity, yet these 'realities' are worth pursuing. If a territory is improperly contextualised, then the project is unreal; it does not represent a purpose. The solution and the problem domain become—unlike the machine-world solution offered by traditional requirements engineering—indissociable. The mapping of the problem becomes reality, and the solution represents a frozen map of its evolution incapable of adaptation. The solution, if frozen, stays helplessly local without a mapping process to accompany it.

However, a general solution from a local mapping runs the risk of detachment. The precise territories from which requirements are sourced are threatened to lose that unique 'locality' character that makes the local the locus of itself. In other words, there is a dangerous disconnect between the reality of the local, and the reality *in-becoming*. The familiar, local terrain, becomes subsumed by forces inconsiderate of that unique 'locality'. The result is a decontextualisation of requirements, generally useful, but locally questionable. The board member, later interviewed speaks of this when arguing for cooperating contextualisations: "you have national repositories [. . .] you have some kind of governance framework around the numbering and cataloguing [. . .] and you have an ecosystem of publishers. You need a new kind of governance which recognises the curation, the librarianship, the skills". The assemblages necessary to openEHR are spelled out: "librarianship", "ecosystem of publishers", "national repositories". The contextualisation has to be curated, he argues, it has to be both generalisable and specifiable by creating specific assemblages of forces that curate their evolution. In other words, to the board member, the territory can only make sense in *both* the local and the global sense, both being necessarily relative to each other and to time.

In the quotation above, the board member distinguishes the locality of the territory, underlining the disconnect problem that can occur. His solution, though, is not to overpower one aspect of territory over another one. Instead, he invokes a monstrous idea: to force the two levels to cooperate through the delegation of responsibilities. To do so, he uses the “librarianship” analogy and the experience incurred in that sector that is, suddenly, not so distant any more. The board member has not switched between realities to the press of a button, but he has bridged territory and its evaluation (which is part of the project’s identity)—two distinct dimensional aspects of requirements—to redistribute control and responsibility in the project’s governance. In other words, reality is not created solely at different levels of locality, but in qualitatively distinct dimensions which come together at a middle. Requirements form one of such middles, which control of is closely associated to the territories and the way these are sourced. In consequence, controlling requirements, and where these are taken from, shapes contexts (and thus, realities).

7.2.2 Ownership and *in-becoming*

If the contextualisation of requirements should have this much power—to shape the project’s reality—then it should follow that whoever controls that contextualisation, controls requirements, and thus, the project’s reality. If this affirmation were to hold, then the project’s becoming would be severely limited, bounded inside frames. In that case, openEHR would only be that much more open than closed projects which frame their requirements early on, deciding unilaterally which requirements are worthy candidates. As it stands, openEHR is challenged openly by open source. This line of questioning results in a clarification of control and ownership over requirements and the sources for the contextualisation of territories.

Open source mitigates the ownership of the project’s requirement sources. A participant argues in favour of a license which does not impede other organisations to use openEHR and start their own competing project. Allowing more or less freedom in the distribution and modification of specifications and artefacts, according to Sundvall, avoids losing any potential territory that could be sourced. By controlling the license, openEHR can choose to control how open requirements are to the world, and in turn, how the world is open to openEHR's sourcing. If openEHR were to put unwarranted limits to the accessibility and distribution of its requirements, it could force other actors to reproduce much of the existing repositories:

> "As I mentioned above it would be inappropriate to try to impose certain licenses on the artifacts/files created using the openEHR specifications, on the other hand the openEHR foundation could of course impose certain licenses in order to use the foundations authoring, validation and distribution services. So yes a CC-BY-SA would be a possible requirement for using the services, but what value would it add in addition to CC-BY? Does it not only increase the risk of pushing organisations that dislike (or that suspect that their contractors/system vendors would dislike) the SA-part to start their own repositories using e.g. CC-BY instead?"
>
> Erik Sundvall, '[openEHR-announce] Interim Statement on Copyright and Licensing of Archetypes', clinical list, October 2009

Another part of this unwarranted reproduction of existing requirements is how it impacts the potential for innovation by yet unknown actors. New territories are limited to the existing political will in openEHR (from its community, to the Board) to help them participate in potential innovation. By removing the possibility for potential users to appropriate the project's requirements and contribute in their own way, some participants worry about stalling innovation and the loss of useful contextualisations. Shannon, another participant, puts this in the perspective of the "front-line" of openEHR, and worries that, through a cumbersome governance model, the project will become unrealistically decontextualised. To maintain that anchor to the "coalface", Shannon discusses the necessary processes that must be maintained for requirements to be contextual. Shannon argues for the freedom of local, opportunistic requirements sourcing to reflect the actual practices at the implementation level. He links requirements to innovation, and having an open license for him is key to allow this bottom-up process of requirements articulation. For the local, familiar territories, he sees a necessary empowerment of small, vernacular and unfettered assemblages.

> "On the other hand, as you and others rightly point out, innovation must equally come from bottom-up solutions, particularly directed at supporting those clinical needs at the "coalface". You may be aware of my drive in recent months to focus some of our efforts within openEHR at the frontline via 2 routes 1)The push for an open source clinical reference framework and application i.e. Opereffa (though in its very early stages) is aimed at frontline clinical use, e.g. SOAP noting. > `http://opereffa.chime.ucl.ac.uk`"
>
> [...]
>
> "So moving on from the need to change and innovate to the subject of "locks" and IP... I would not/will not support any attempt by the openEHR Foundation to "lock up the IP" in a way that gets in the way of clinical innovation and improvement in this space. If anyone is concerned that the interim statement is getting in the way, please let me know. My understanding of this move towards Creative Commons licensing is quite the opposite to be honest...(more on that shortly)"
>
> Tony Shannon, '[openEHR-announce] Interim Statement on Copyright and Licensing of Archetypes', clinical list, October 2009

By qualifying its degree of open, the problem open source poses here is not only the loss of sourcing territories; it is also a question of maintaining interoperability within the project. The license choice, is both dependent and responsible for the sourcing and articulation of other requirements that can help sustain interoperability in the foreseeable future.

7.3 Conclusion

It seems improbable that territories could be a cause of instability—for good or otherwise. Territories have been mentioned in issues of unframed boundaries, unclear project objectives, disconnect, innovation, control and ownership, fragmentation and protection. Traditional engineering tries to over-power territories by defining unilaterally what is 'in' and what is 'out'.

openEHR shows territories to be luxuriant with controversies, with far reaching assemblages. They directly touch meta-variables such as time, locality, contextualisation, qualities and identity. They act as the soil between the perpetually immobile (and the efforts that it requires), and the naturally evolving *Being* of requirements. Rhizomatic objects, requirements use their potential to take root in 'foreign' soils, evolving both in substance and expression to be *apprivoisés*, adopted into new contexts. To source and articulate requirements, openEHR needs to be present and active in those territories. Despite—or because—it is an open source project, it has to deal with ways in which it can mediate and derive new requirements from explored and unexplored territories alike. These territories are often inconspicuous and emerge somewhat spontaneously, calling the project to provide its attention and cast its light. What becomes seen by openEHR, thus, depends on its ability to deploy and coordinate resources (while protecting its identity and governance model). The front-line, for example, avid of recognition, pushed for ways to become immediately useful to themselves and the project as a whole. The way, thus, these territories are apprehended depend on how the project is capable to engage requirements and integrate them.

Chapter 8

openEHR Engages Requirements

This chapter discusses requirements as a middle. While previous chapters have looked at the assemblages that ally with requirements, this chapter focuses on requirements' forces and the assemblages *they* design. There is a balance to be struck, a middle, so to speak, in the forces and assemblages that requirements produce and draw. The forces they get are partly due to their own nature, to their substance, and the forces that the political processes imbue them with. These political processes are far from being obvious, and requirements are often articulated through a jungle of ambiguous meanings and misunderstood consequences. The result are requirements with multiple forces, yet barely understood ones. To manage that multiplicity of potential, certain modes of engagement are employed, often paradoxically, together. Such engagement applies some qualities to the requirements, indissociable to the sketched requirements. Requirements quality could be said to draw together both the substance (the material), and the expression (the expressive force, materialisation). The result is an intricate play between substance, expression, and the potential of becoming.

This chapter analyses:

Modes of engagement. A requirements' mode of engagement can be through emergence and stabilisation, which involves a careful balancing act to benefit from a requirement's becoming.

This is not a requirement. Requirements hold substance and are transformative in their expressions. There is no complete requirement (ever). The way requirements are given substance and articulated also influences how further requirements, and the project itself, can express its desires.

8.1 Modes of Engagement

There are two main modes with which openEHR engages requirements: through their stabilisation or/and their emergence. These two modes are 'modes' in the sense that participants try to evaluate and attribute values to requirements to their ontological beliefs. The title of this section, 'Modes of Engagement', is borrowed from Latour's ontological work on 'Modes of Existence' (Latour 2013). These modes express the intent of many actors (among which are requirements) to engage, according to a certain logic, with openEHR's requirements. For example, when a participant talked about creating requirements through "rigour and engagement", he was not only referring to a requirements' own qualities, but the overall process and the frames of mind in which they are created. The frame of minds are important to the articulation of a requirement's qualities (e.g. it represents the world accurately), and are also responsible for the way the processes of articulation are evaluated by the participants.

Engaging requirements through stability is about rooting the requirements, exerting increased control on their features and representation, to the risk of diminishing its potential to change. Emergence, on the contrary, leaves more room to requirements to gain more personality and create assemblages; increasing their potential to follow their desires and become other. Each mode express requirements within their own logics. This section analyses these two modes and examines their various consequences.

8.1.1 Emergence is King

The link between emergence and improvisation is a tight one: "Looking at improvisation as a special disposition or attunement with the situation, a special way of being amidst the world and being thrown into it, opens up a different point of access to the phenomenon: improvisation as mood" (Ciborra 2002b, p.162). Situation is an important, localised, element which affects the way emergence is dealt with. Taking a larger scope to emergence, seeing it as the "mood" Ciborra speaks of, emergence becomes a 'mode' and concerns the forces that are put into place to care for the emergence of requirements.[1] These forces touch the emergent requirements' affects and their potential actions: are emergent requirements accepted unchanged into openEHR? What are considered good emergent qualities for requirements? Is the governance adequate to protect emergent qualities? These are some of the questions openEHR deals with when discussing the emergence of requirements.

More than local, situated action, it is time which seems to be a specific quality in the emergence of requirements in openEHR: no artificial boundaries are placed to *when* a requirement is (making the *what* that much more difficult). The project does not go through only one, or limits itself to a few, requirements 'capture and define' moment the way iterative or even Agile methods would do. Emergence is supported by an open source approach of 'plastic', pliable requirements, which openEHR illustrates by its willingness to invite adapting openEHR to any specific use *as long as certain rules are respected*.[2] In other words, the plasticity of requirements is guided and modelled according to those rules. This might seem like a paradox that is worth exploring further.

It is typical to find questions by potential participants asking about openEHR's interest in a particular requirement they are interested in developing:

[1]Some manifestations of these forces, such as licenses and values, have already been analysed in previous sections: they also have a hand in collectively articulating the requirements' expressive force and substance, though they may remain invisible to local situations. They also help in its sourcing by allowing anyone technically or clinically knowledgeable to formalise their own requirements in the project.

[2]Even then, the possibility of forking open source projects gives a wide variety of evolutionary possibilities.

> "I'm trying to archetype patient records in a prostate cancer environment. I see that there are currently no specific archetypes for common cancer treatments such as radiotherapy, hormone therapy and chemotherapy. Could hormone therapy and chemotherapy be described as specializations of the openEHR-EHR-INSTRUCTION.medication.v1 archetype? And could radiotherapy be described with a specialization of openEHR-EHR-INSTRUCTION.non_drug_therapy.v1? Or should these be described in completely new archetypes (e.g. INSTRUCTION.radiotherapy/ hormone_therapy/chemotherapy)?"
>
> Melanie Spath, 'Cancer treatment archetypes', clinical list, November 2009

This kind of contribution is always welcomed:

> "That is really helpful. If you and Melanie do have some discussions it would be really useful to capture some of this knowledge around radiotherapy, even via something a bit rough and ready like a mindmap.
>
> I am hoping Rong will chip in with his experience of chemotherapy guideline modeling, which, of course, mirrors the actual prescribing instructions.
>
> Would hormonal therapy always be akin to a medication order?"
>
> Ian McNicoll, 'Cancer treatment archetypes', clinical list, November 2009

Often, some resources and knowledge is shared to precise the requirement and help in its development, even though as is the case in this thread, it might be a long-term project:

> "We have also just completed a project for an Australian cancer epidemiology center where we converted all their 20 years of data to openEHR - about 1.5 million compositions and 25GB of data. To do this we had to do a lot of modeling (although in the end, they did most of that themselves) and that experience and some of the archetypes that came out of it might be of interest to you. They are now doing poplulation queries across the whole dataset using AQL. Very exciting!"
>
> Hugh Leslie, 'Cancer treatment archetypes', clinical list, November 2009

There is no prior contact with Spath to originate this requirement. Something in openEHR is attracting participants to contribute. In part, it is the existing functionality of openEHR such as its philosophy, the alignment of needs between the participants and openEHR, and also the needs' potential to become. If this *intéressement* is at all possible, it is because of openEHR's *openness* in the sourcing and articulation of requirements.[3] openEHR, through the materiality of its license, creates prior conditions that allows the emergent engagement of unexpected participants.[4]

The concern for emergence is found throughout discussions that touch the process of developing requirements and standards making. Participants recognise anyone's incapacity to understand all requirements, especially compared to the available resources to source and articulate them:

> "[. . .] everyone is asking for what they need, which is way smaller than the total demand, and this is mostly likely to be the reason for people to say "how hard can it be?, I'm just asking for XYZ!" Delivering what a party asks for, without breaking the consistency of the solution (which makes it a solution in the first place), requires a lot of work and coordination.
>
> [. . .]
>
> Just like many other groups out there, openEHR is suffering from an asymetry. The input regarding the requirements and what should exist is gigantic, compared to input to deliver the results. Also, the cost of making a request is much lower than actually responding to that request.. This is not a bad thing, not a complain or rant, this is just a fact of this kind of organization. It is just that you need to acknowledge this situation to solve the problem, and develop a way to solve the problem with this picture in mind."
>
> Seref Arikan, 'Why is OpenEHR adoption so slow?', technical list, November 2009

[3]Whether this openness is open source or not is a debate in openEHR. In some aspects, openEHR has embraced open source as Callon would describe as an obligatory passage point for anchored, interoperable, transparent requirements (among other qualities)(Callon, Lascoumes, et al. 2011). In other aspects, openEHR is still trying to apprehend open source and answer complex questions: how it should be used to create these requirements qualities and allow a diverse community to contribute adhering to some strict rules to maintain those requirements qualities?

[4]This materiality is debated by some participants and further explored in the discussion chapter when materiality and materialisation are compared to forms of expression and substance.

Arikan highlights a double-edged sword for the emergence of requirements in open source. On one hand, open source is able to source an incommensurate amount of requirements, but on the other hand, openEHR has limited articulation power to mould them into openEHR requirements rapidly enough. The early benefit of not framing off requirements artificially (as would traditional process of requirements) and allowing emergent requirements poses a serious organisational difficulty that openEHR must manage. Managing this organisational issue is not evident, and participants argue for various tactics and strategies to administer. Again, the issue of control seems to be as problematic as it is useful. When it is useful, it is met with stupendous words:

> "It's not coordinated, it can't be, because it's all left-field stuff. So GPSoC I knew nothing about.[5] And I mean quite honestly that's the way I would want it to be because I think a thousand blossoms blooming is really the nature of where we're at, at the moment, because we are not going to know exactly where the whole thing will resonate and where it will add value. We had no idea that somebody in Cambodia was going to download Opereffa [*an openEHR demonstrator*] and build a TB [tuberculosis] national alert system."
>
> An openEHR Board Member, interview

At the same time as presenting a problem, control—with the aid of emergence—seems to solve it. The emergent requirements which fit well with openEHR help contribute to openEHR's sourcing and articulation of further requirements. Those emergent requirements that do not fit openEHR help propagate openEHR's identity into more territories, ensuring the project evolves and does not stagnate. The non-fitting emergent requirement, even if non-core to the project, will diffuse openEHR's paradigm of archetypes—the requirements featuring in projects similar to openEHR.

The emergence of requirements, and the dilution of their 'ownership' has a consequence on coordination. There is coordination, but it is not entirely centralised

[5]The openGPSoC (open General Purpose Source Custodian) is an umbrella project that aims to use openEHR. openGPSoC was launched at the NHS Hack Day 2 in Liverpool (a grassroot movement which aims to 'hack' the NHS. A test build is currently being developed.)

into one head which would control functions. Rather, groups and communities form that respond in their own way to the centrality that openEHR—as the main actor—projects. In other words, these groups and communities will translate in terms of requirements, what, for them, openEHR represents.

Is this a bad thing? Should it be considered as a challenge to the core? In many ways, it is. openEHR's power is diminished from it. Yet, its centrality is also reinforced. It is a paradox that an entity could lose power, and yet exert more force as a consequence. In other words, openEHR cannot stop any other entity from using its requirements in ways it does not agree to (unless it goes against the terms of an open licence, which leaves a wide berth of movement), but its role, its purpose, its function beyond that which is written in the codes, the norms, and the licenses, is increased. It does not dictate behaviour, but it guides the blooming of "blossoms" by its work and through the emergence of acceptance of potential *détournements*. The purpose of openEHR overflows (Callon, Lascoumes, et al. 2011).

One consequence of this renewed centrality is that openEHR has an important role of guidance that it cannot shy from, at the risk of undermining its force. This can be seen as a negative side of emergent requirements which can disrupt openEHR.

> "I do have one negative comment and I'll get that out of the way first. As I have indicated, I believe that I will have more success working on some of the fringe areas of healthcare with openEHR. Therefore I tend to talk to providers that are not in the main; primary care/family medicine/general practice areas. I have been told by more than one of these folks that they didn't feel very welcome to participate on this(Clinical) list on issues that concerned their areas."
>
> Tim Cook, 'Wisdom of the Crowds', clinical list, February 2009

Cook himself had stopped being an active contributor to openEHR and went on to create his own organisation, using aspects of openEHR, but under his own governance. Still, the openness of the project allows him to participate in discussions and even take part in board meetings. In this sense, Cook 'scratched his own itch', while remaining attached to openEHR.

8.1.2 Wary Opportunism

Just as a king is absolute only relative to being accepted as such, emerging requirements in open source are not the only source of law. As informal as research depicts open source projects, openEHR exhibits a concern for stability that contradicts this informal by default view. If emerging requirements are seen as positive contributions to openEHR, how the emerging forces are expressed and given substance is an issue. The mindfulness behind this stabilisation is an evaluative force which imbues certain qualities into the requirement's substance.

The emergence-to-be-stabilised is qualified: there is a right and a wrong one (and various shades of grey). These judgements depend on a variety of factors. The interoperability requirement, for example, is in tension with almost every other requirement. It takes its substance from the collaboration of other requirements, entering a symbiotic alliance where each can express themselves in mutual force, or erode each other. To help maintain interoperability (among other requirements), a board member speaks of stabilisation through rigour:

> "[. . .] but I mean they [requirements] have been revisited all the while in the context, and what I have tried to insist on is that whenever we do things at the architecture level we pin them back on requirements. So, you know, we need to say, this facet of the way things are structured and organised reflects the way that we've approached these requirements, and we can always pin things back. Because otherwise, you make changes which are purely opportunistic, you know, it's easiest to fix this, this, and this to achieve that limited, short-term goal, and if you do then, you quickly lose the thread, the rationale. So it's all about the discipline of, of, and the rigour of it really.
>
> [. . .]
>
> Also, the rhetoric of it is fine, in a sense, not fine because it doesn't implement but, that's a lot of the fundamental way of doing it, you know, need much more experimental rigour, but the problem is in this sort of contexts, good people end up doing their own thing, because they feel they have no choice. So that's what happens with local requirements, they then fracture the enterprise and then health care as a whole is lost, because actually that doesn't help, we end up in the middle of UCLH, hundreds of different clinical IT systems, none of which has got any common basis and standardisation of data or anything else, all of which the developers have long since

> moved on somewhere else, and wax holding it together."
>
> openEHR board member

Rigour, here, is part of the mode of stabilisation. It is an evaluation *and* a way to approach requirements. It is also situated action: the board member speaks of "experimental rigour", "contexts", and to "pin things back". Emergence and improvisation do not have a monopoly over local actions and a strange—uneasy perhaps—alliance is made that aims to transform opportunistic requirements into acceptable, even adoptable requirements. This mode, then, is as much a process as a qualification of what a good requirement is. It opens or closes openEHR's doors to a certain type of participation, in accord to the wider principles of the project and the other requirements and actors with which interoperability is assembled with. The board member seems to say: requirements must become one of us.

But the board member is not speaking about the expressive force of the requirement, its articulative expression. There is no overall appreciation of what a good requirement should do, but how it should be. It is the substance of things he is worried about, whether these timbers that should hold the project for the future years fits with the already gathered assemblage. Functionally, openEHR might be heterogeneous—becoming a tuberculosis warning system in Cambodia—but in substance, it should remain homogeneous. Is this not a rhizome? The homogeneity is a plastic identity, recognisable in its essence, but expressible in different contexts, depending on the needs.

> "Remember I've coined this phrase of rigour, engagement and trust. And the rigour bit, for me, in the scientific world, most of physics couldn't exist without open source software, because that's the way people know, you know, software is extraordinarily complex, unless you've actually got it in your hand and you work with it, you don't really know, and there's so much software around in the world that nobody really knows what."
>
> openEHR board member

The board member is not against opportunistic requirements, but against their emergent ‘impurity’, which, if not transformed could make the requirement timelessly emergent. In essence, to stabilise a requirement is to exchange a quality—to make a cut, so to speak—to the requirement so that it may assemble with other requirements. To openEHR, to rearrange this (im-)pure, local, opportunistic requirement is to exchange an ever-emergent quality for an assemblage (e.g. the project). In that assemblage, the requirement could become something else or help openEHR *in-becoming*, but one type of time is exchanged for another one. The requirement’s time is not its own any more, but the new assemblages it is in. Time, as Deleuze said, “is constituted only in the originary synthesis which operates on the repetition of instants” (Deleuze 2004, p.91). In other words, time is the (re-)affirmation of the repetition. If too many repetitions are adopted into a project without any transformation, what ensues is an amalgamated, hierarchical project with a composition of identities. In other words, a false multiplicity (Deleuze and Guattari 1987). The project’s becoming, thus, would be lessened. Therefore, integrating requirements is one way of driving transformations, but only if the exchange—time, potential, space, values—is done appropriately: the requirement has to change when coming into the project, and the project also has to accommodate to the requirement. In other words, it is not only the requirement alone which must undergo change, but the project as well.

There is an element of proximity creeping behind the shadows in the emergence of requirements. The “opportunistic changes” mentioned by the board member refer to localised requirements. These requirements are then de-localised by rigour-stabilisation. But rigour-stabilisation has no intention of uprooting. There is a sense of proximity that the board member wishes to preserve otherwise “you quickly lose the thread, the rationale”. The good requirement, then, is rigorous, being both anchored and the result of a systematic process of alignment with the project and the global environment. To engage in any meaningful way in openEHR is to do so through a rigorous experimental process that would take, for example, an opportunistic requirement, and ensure a rational between its existence and the other requirements so that more than “wax” holds the requirements together.

To do this moulding of requirements according to openEHR's values, participants need tools and processes. Arikan mentions some of them:

> "We need to do a huge amount of work, and I personally don't see this work being done in any other way than a properly funded, planned, and managed approach. You can't break down all tasks and diffuse it into some good intention based completely democratic virtual work force. openEHR has lots of tasks with this nature at hand, and many things which has worked in other scenarios won't work here because of this."
>
> Seref Arikan, 'Why is OpenEHR adoption so slow?', technical list, November 2010

Arikan's email shows the limits of the emergence of requirements by limiting openEHR's capacity to accommodate them. There is another reality that supersedes the requirement's qualities and value and which also limits the project's potential of becoming. If emergence is to be useful, it needs to be realised within openEHR's capacity to grasp and accommodate the requirements. Emergence cannot be on its own, and neither can stability.

openEHR displays an entangled relation between emergence and stability. The greater the distance between them, the less sense they make. Emergence cannot be represented as the opposite of stability, and vice-versa, because both are immanent to a requirement's potential. If emergence or stabilisation were to be taken away from requirements' becoming, requirements would lose meaning relative to their assemblages (e.g. the project, the front-line, the other requirements, etc.).

Deleuze makes much the same point when he says: "What is immanence? A life… No ones has described what *a* life is better than Charles Dickens, if we take the indefinite article as an index of the transcendental. A disreputable man, a rogue, held in contempt by everyone, is found as he lies dying. Suddenly, those taking care of him manifest an eagerness, respect, even love, for his slightest sign of life. Everybody bustles about to save him, to the point where, in his deepest coma, this wicked man himself sense something soft and sweet penetrating him. But to the degree that he comes back to life, his saviours turn colder, and he becomes once again mean and crude" (Deleuze 2005b, p.28). Good and bad here, are not relative (they are not given), but relativistic; they feed off each other. So do emergence and stabilisation. Emergence is often local, but desires grandeur skies, while stability is emancipated, but can overlook the local.

8.2 This is not a Requirement

The painting by Matisse of an apple and a man's face, is evocative for requirements.[6] Which is the apple in the Matisse painting? How can an apple be recognised for what it is? Certainly, the context plays its part, which is why Matisse calls his painting *"Ceci n'est pas une pomme"* (this is *not* an apple); the negative serves to show the absurdity of the apple's placement, or the man's mysterious apple-face. Yet, this *double-entente* is easy to understand in the case of an apple, but more difficult in the case of requirements.

If the previous sections have looked at the various ways requirements can shape and describe reality, this section analyses requirements themselves. It asks what could be and what could not be a requirement, and for there to be a requirement at all, there needs to be an identifying substance, a common *je ne sais quoi* that accompanies their meaning, even if that meaning were to be a multiplicity. In other words, there needs to be an element of immanence, a flexible enough material that allows requirements to become multiple things of varying intensities. Like Matisse, this section is not really interested in what is or what is not a requirement, but instead on the meaning they can hold and come to mean.

Concretely, this section analyses:

Substantial requirements. The materiality of requirements.

From properties to functions. The materialisation of requirements.

[6] *La pomme* (apple in French) can mean both the fruit and, colloquially, someone's face.

8.2.1 Substantial Requirements

In openEHR, requirements are explicitly composed of substance. The substance itself is a force, it resists and allows change, and accommodates further expressive articulation. openEHR reveals aspects of requirements' materiality and substance. In a request for reviewers to "adopt" the 'Adverse Reaction' archetype, the requirement is weighted as particularly "significant", requiring "broader expertise" from the community:

> "A review round for the Adverse Reaction archetype was initiated today. This is a significant archetype that requires careful and considered collaboration and I would like to ensure that we have the best team reviewing the specs as we can.
>
> Current reviewers comprise the openEHR Archetype Editorial Group plus existing adopters of the archetype, however we welcome broader expertise from the broader openEHR community."
>
> Heather Leslie, 'Adverse Reaction archetype - review round initiated', clinical list, July 2009

This requirement is not innocent to some people. Personal motivations come and mingle with territories and assemblage space over what requirements should have responsibility over. The various shapes of the requirements and their assemblage's substance appears in Grieve's email: should it be "a report" over a direct observation, thus carrying greater implications? Should it be a "high level clinical summary" to help future decisions? The substance as it stands, Grieve argues, is "ambiguous":

> "I'd like to start by asking about the scope of this archetype. Is the intent to report about a condition of an adverse reaction? or a concern about? or just a report of a possible one?
>
> Is it a high level clinical summary, or is it supposed to be good enough to support DSS [*Decision Supprt Systems*]?
>
> I have considerable interest in this archetype: as well as being involved with this model through NEHTA [*National E-Health Transition Authority for Australia, follows openEHR's paradigm*] and HL7, my daughter is highly allergic to tree nuts, but (a little unusually), not peanuts as well.
>
> It seems to me that the current archetype is only good for a gross point report of a single episode of apparent adverse reaction.

> If this is all it's supposed to be, I won't have much to say, but if it's supposed to be good for more than that..
>
> I'd like the archetype to comment on this. "Recording the presence of a harmful or undesirable response to an agent or substance including food, as determined by the clinician – excluding poisoning and abnormal use" is ambiguous concerning these questions."
>
> Graham Grieve, 'Adverse Reaction archetype - review round initiated', clinical list, July 2009

Reports, clinical summaries, direct observation, are all concepts that requirements could come to *represent*, partly through their expressive force (e.g. the representation of a direct observation of an adverse reaction), and partly through its substantive force (e.g. the space and assemblage the requirement fits in). Lövström contends that the representation of an 'adverse reaction' should fit "a certain point of clinical care (e.g. prescribing, feeding, anaesthesia)":

> "To some extent I think it all comes down to being aware of (recorded or feared) risk factors (reactions to a specific agent or conditions and treatments which could give equally serious reactions) at a certain point of clinical care (e.g. prescribing, feeding, anesthesia) where the clinical acting could be altered from a 'standard procedure' to avoid a risk of negative conequences for the patient.
>
> To start with, maybe just separate the cases you talked about? Recording an apparent reaction, recording a possible one (could be the first one with less degree of certainty) and then looking at recording risk factors like conditions and treatments later?"
>
> Rikard Lövström, 'Adverse Reaction archetype - review round initiated', clinical list, July 2009

Lövström, by considering the requirement's appropriate place in the business environment, establishes a link between a representation of reality *and* a constructed substance for the 'adverse reaction' requirement. Discussing a requirements' space disputes the traditional view which would consider them ethereal forces or immature concepts needing parental guidance to be expressed. Instead, Lövström points to complex relations and assemblages: requirements are actors purposefully engaged in disjoining an apparent unified reality into a new, *in-becoming* one. If the requirement is not in a correct situation there is a "risk of negative consequences for the patient". It could well be that all the ambition behind the requirement is for nothing, that Lövström's argument is inconsequential. Yet, the requirement and its assemblages hold the potential to occupy a place in the world and affect it through its material substance and expressive force. A nurse might have to indicate which moment an observation was made, a doctor might use the information contained, and the materialisation of the requirement into a force which leads to a decision about a treatment. A doctor might wonder, just as Lövström did, whether the recording was an "apparent reaction", or "a possible one". What should the doctor think if it was not explicitly possible to emphasise the degree of observation. How would the requirement make invisible or uncertain parts of a doctor's practice? Requirements, *through* their substance are both representations of reality, and a potential reality. Substance and expression are inextricably related.

Lövström's articulation of a requirements' aspects illustrates how they can stand on their own, being both an expression and a body. Lövström is not inventing the requirement. Neither is he chaperoning it. Instead, he uncovers the requirement's complexity, revealing problematic aspects that its inappropriate existence could create. If a requirement were entirely definable, then its unintended consequences would be too. In other words, a requirement misbehaves from the point of view of their creators precisely because the creators have never had control over it (at least not an over-powering kind). A requirement holds its own potential place in the world and influences it through affects of its own; and sometimes these are hard for actors to understand because they have no easy 'handle' to render plain the consequences.

Time is again of the essence. Leslie makes the link between the requirements' substance and time clear: "the current Adverse Reaction archetype is a 'straw man' model". To simply be a "straw man" requirement involves a great deal of work and consensus that should only be refined into specific requirements at a later time. The requirement, "straw man" as it may, must be of that particular nature to have the transformative force to enable further refinement later. As Leslie reasons, the requirement serves as "the basis for sharing common things about adverse reactions". Leslie points out the importance of the requirement's substance in time, and its evolutive forms which should be applied to more contextual environments. Leslie's efforts aim to produce a requirement that holds potential to evolve into yet unknown contexts. Specific cases should be attended to at another time:

> "The current scope for Adverse Reaction is the initial ideal for every archetype at creation - a maximal dataset for a universal usecase.
>
> So the current intent of use of the archetype includes within a report, within DSS [*Decision Support System*], a clinical summary and any other way of utilising it. The archetype can then be constrained to make it 'fit for purpose' through context/scenario-appropriate templates.
>
> The current Adverse Reaction archetype is a 'straw man' model that needs considerable refining and enhancing - no doubt about it. It represents the thinking of a few people, based on lots of experience and only one reference noted. The purpose of this review is not to gather yet more business requirements but to collate the existing research and thinking done by many learned and expert organisations and national programs into a practical and pragmatic model that we can take forward as the basis for sharing common things about adverse reactions.
>
> As you quite rightly point out, the metadata needs quite a bit of enhancement. The review process should identify any missing data elements, and usecases not currently anticipated. The metadata will be improved to support all of this. "
>
> Heather Leslie, 'Adverse Reaction archetype - review round initiated', clinical list, July 2009

All the while, the requirements process is in the background, seemingly lifeless. The only indication that it is around is the original poster's email requesting participation on the review round. In this discussion, the process holds no life-giving status, it is no creator here. When it comes to the foreground, it is as a support to the requirement more than an engineer's framing process:

> "Do it on CKM - not on the list! Then the ideas will not be lost. The proposal certainly covers more than you have noted but would not in itself support a report. This would be a template."

> Sam Heard, 'Adverse Reaction archetype - review round initiated', clinical list, July 2009

Heard's call signals the subjected position the process holds here: it is there to record the requirement's substance and expressive force, not to pilot anything. The reviewing process is helpful to the requirement's substantiation. The scope of the review depends specifically on the requirement's ambition to have an "initial idea for every archetype at creation—a maximal dataset for a universal use case". This condition is neither procedural nor intrinsic to the requirement. It is part of another requirement's force, a particularly forceful one: interoperability, and the assemblages it has knitted. This force has been part of openEHR's identity for a long time. It has, in fact, guided its steps alongside the project's creation:

> "[openEHR will:] offer all its work openly in a spirit of a public enterprise, believing that this is the best and perhaps only way that appropriate high quality and interoperable systems are likely to emerge, worldwide."
>
> David Ingram, Foundation board president, 2002[7]

Requirements are not born in a vacuum since they are, even before their sourcing, influenced by other requirements and assemblages. The requirement for interoperability has become such a force that, years later, still influences the sourcing and articulation of new ones. Often, requirements are defined by their apparent properties. The 'adverse reaction' archetype, for example, could be said to be general, significant, consensual. One of its function under discussion: "to report about a condition of an adverse reaction", is legitimised by the properties it is attributed with. But in so doing, the function itself is not treated on its own ground. the requirement is said to *be* that function, to describe and embody it. The function, the requirement, and the properties become merged in an assemblage that is only disentangled when the requirement is articulated, or controverted.

[7] http://www.openehr.org/about/origins.

8.2.2 Expressive Requirements

Requirements are often discussed in the openEHR mailing lists. Sometimes, these discussions are about a specific requirement such as the specifications of CQuantityItem units, at other times, general perspectives are shared. Most times, though, discussions often range from the general to the specific, where a general idea will be contextualised, or a specific requirement will be generalised. These contextualisations and generalisations are not without consequence. In their framing, they carry a surprising amount of becoming about the *meaning* of requirements: it is often the case that clarifications about aspects of requirements are necessary. For example, the role of archetypes appears to be ambivalent for some, until its purpose and responsibility is made clear (and sometimes needs to be 'remade' several times). This gives a strange sense that requirements derive in part their expressiveness from their substance. The rational for requirements is often necessary to understand the meaning of a requirement. In an email to the clinical list, Pazos asks this:

> "I'm playing around with archetypes trying to model an observation and its reference ranges, I mean something like "blood pressure" and some range to define what is "hypertension", but I can't found an archetype that defines a reference range for an observation.
>
> Any one has experience in modeling something like this? An archetype is the correct place to define a reference range for an observation value? Any ideas?"
>
> Pablo Pazos, 'Modeling reference ranges', clinical list, October 2009

Pazos's email indicates the degree of 'plasticity' of a requirement, and at the same time, the strictness of its malleability. To model this requirement, Pazos has had to "play" around with it since he did not find any obvious guidance to describe what he wanted. He also has to ask anyone with "experience" to help him find the correct place for such functionality. Functionality is not evident, and sometimes demands an understanding of the rational behind the requirement:

> "The Quantity datatype in the Reference model has built-in support for Reference ranges so these do have to be modelled overtly in archetypes.
>
> See `http://www.openehr.org/svn/specification/TRUNK/publishing/architecture/rm/data_types_im.pdf`
>
> This makes sense for lab tests etc where each test will report a reference range (which are often lab/analysis method dependent) along with the results themselves.
>
> However, you are talking about something different. There is really no such thing as a reference range for blood pressure, which might indicate hypertension. The definitions of hypertension vary, over time and by locality and the diagnosis will depend on many other factors than just the blood pressure itself.
>
> I think what you may be trying to capture is some thing more like a 'trigger blood pressure' which displays an alert to the clinician or initiates some other action if a set of criteria have been reached e.g 3 readings with a diasstolic > 100.
>
> This is more akin to a guideline or care pathway."
>
> Ian McNicoll, 'Modeling reference ranges', clinical list, October 2009

McNicoll shows the limiting view of equivocating requirements and functionality. There is a more meaningful way to model reference ranges. A requirement's functionality depends on the requirements' needs. In other words, specific choices are made for requirements which articulate other requirements further down the line:

> "We do not set the reference ranges in archetypes as these vary and archetypes are the absolute statement about things (what could possibly be true ever, anywhere)."
>
> Ian McNicoll, 'Modeling reference ranges', clinical list, October 2009

Archetypes could have been made to record reference ranges, but this would have gone against the values and identity of what archetypes are and should become. The usefulness of such a requirement is put in perspective with the purity of the requirements design:

> "Thanks Sam,
> That was helpful but would you agree that is does not make much sense to use a reference range for blood pressure in the same manner as you would for a lab test. I have suggested that if Pablo is trying to set trigger conditions e;g a series of BPs over a particular level, then this properly belongs in the guideline/pathway space, rather than as ref ranges?"
>
> Ian McNicoll, 'Modeling reference ranges', clinical list, October 2009

Reality forces the use of triggers on data, and shows the importance of requirements in revealing, in turn, aspects of reality:

> "I agree Ian - though they are always triggers in reality. Australia made more progress on Lipids when it changed labs from reporting actual norms to target norms. Suddenly everyone had high cholesterols and down they have come!"
>
> Sam Heard, 'Modeling reference ranges', clinical list, October 2009

A requirement's expressive force, thus, in contrast to functionality, concerns the requirements' ability to shape reality. It has a certain weight that pulls projects and other requirements around its mass of influence. If a requirement's functionality tries to describe a software's reaction, the requirement itself influences and is influenced by other actors and assemblages. A requirement's materiality is itself a living force: a requirement representing "target norms", according to Heard, would hold greater sway over actors, just like enforced norms by Australian institutions. Requirements, then, have a greater responsibility in an organisation's goal attainment than through a narrow functional, machinic description.

Requirements carry a certain weight about them and influence engineering processes, values, and territories to visit. There is a particular thread which exemplifies this: when Arikan asks about the integration of UI technologies and their representation of archetypes, various high-level perspectives of modelling health care are confronted, with the correct articulation and sourcing of requirements as the pivotal apparatus. The question is simple, but deep: can interoperability exist? The various positions imply different postures regarding requirements. It is unclear though, whether these different postures force requirements to be formed in a certain way, or whether certain requirements and the qualities they materialise force these diverging postures. In other words, as Massumi would say (1992), which is the form of substance and which the form of expression? Are requirements enacting their world, or is the world oppressing them into being?

> "Good points. My humble opinion is that, the specifications which are targets of mapping are moving targets, and researchers in the semantic web field seem to be in "art for the sake of art" mode (absolutely no offense). That is, they seem to building capabilities for processing relationships, and processing the actual items involved in relationships is the missing part of the puzzle, or at least that's my holy grail. I ran into Dr. Dipak Kalra today on my way to grab coffee, and he gave me a couple of nice real life examples, about the things he'd expect from a well established system that has semantic interoperability features. Basically as a clinician he is demanding the capability to ask questions in a single form to heteregenous systems (unless I misunderstood), and this is a good example of a use case, where you need access to real life data, therefore crossing the boundaries of semantic web and ehr related research in both ways."
>
> Seref Arikan, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

The problem for openEHR, and generally, for any system that aims interoperability is the gap concerning the requirements that allow for interoperability.

> "I think that there are clinical informaticians that know, implicitly or explicitly, about semantics, about language and the philosophical aspects. At least clinicians and nurses do (and most patients and other people) since they communicate using voice, writings and gestures.
>
> The problem is that technicians do not understand semantic interoperability. And I must say that many informaticians are actually technicians without any understanding of semantics."
>
> Gerard Freriks, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

It is not so much the sourcing of requirements that is called out, but the articulation of that sourcing. In other words, the problematic of what, where, and how, requirements need to be articulated. It is clear that both clinicians and informaticians are needed, but how to share that understanding? This question makes another participant ask, half jokingly, whether "we should aim for unsemantic interoperability." Freriks then proposes a more practical solution where requirements should be defined in a specific space:

> "Dear Seref As a more technical continuation: When ontologies and syntaxes are orthogonal the two meet in one place At that spot on the syntax will refer to a code from a coding system (terminology, classification, code list) Technically it boils down to how semantically correct and safe can we define this reference?
> Ontologies can play a role in the prlduction of codes
> Gerard"
>
> Gerard Freriks, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

It is hard to find that 'spot', where requirements are both meaningful (to clinicians) and expressible (by informaticians). Yet, it exists, almost as a physical space in the minds of the participants where a correct balancing act between raw requirements, and requirements syntactically valid for computation can be made. Others, still, believe in the possibility of creating semantic links between various specifications through formal mappings. The problem with this as Arikan argued, is the same problem that has been afflicting requirements engineering since its inception: the 'ungraspability' of requirements. The formal mappings aim to stabilise otherwise rapidly changing *and* ambivalent or multiply meaningful requirements, with dubious success. It is a problem of sourcing requirements and articulating those sourced requirements. Freriks makes this point clearly by separating two physical worlds: the "models of reality" and the "models of documentation":

> "Mixing two different types of models is impossible. The best that can happen is that in one model-world one refers to constructs

> in the other world."
>
> Gerard Freriks, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

The functions and qualities of requirements are different from a world to another. As Freriks argues, in the models of reality, "codes, descriptions, labels" are modelled to make "inferences, to express knowledge". Models of documentation are for "archiving, exchange and re-use. All data and information stored" to allow "users to write complex sentences, using documentation patterns humans agreed upon."

Despite these physically and spiritually separated requirements, the boundaries are not as clear as they might seem. Another participant argues that this divide is more of a grey area.

> "I agree that there is a difference between language and ontology. I am less convinced that to serve clinical system interoperability the distinction can be maintained absolutely. At one level there is the blurred boundary between terminology and structure, and at another there is the safe automated reuse of entries/clinical statements - something that happens and for which we need a better understanding, with entries being treated as semantically independent. I beleive that ontologists have much to contribute to this area."
>
> Charles McCay, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

The problem for a requirement's expressivity is, again, a physical one. At what level should requirements be sourced and articulated? Participant b.cohen points to problems of standards not meeting criteria and, on the other hand, ontologies being too ambiguous to ever be stabilised:

> "The sole purpose of a standard is to guarantee interoperability but, to achieve this, the standard itself must satisfy certain criteria. [...]
>
> Unfortunately, few, if any, of the 'standards' in Healthcare meet these criteria.

> Ontologies (another term coined by Peirce, who also did the original work on ontology definition, as Sowa himself acknowledges) provide some leverage but, since there can be no universal ontology, and the composition of disparate ontologies is not computationally feasible, only those communities who can identify with a common ontology can benefit, and as this identification is always merely temporary, later dissatisfaction with its implications is inevitable. I hope this is not seen as too pessimistic. There are effective ways of dealing with these problems but only if we learn how to include the subject, and the subject's models, in our models. The 'objective' forms of analysis that have been so successful in engineering simply will not suffice."
>
> b.cohen, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

The space in which requirements are sourced and articulated is constrained in b.cohen's argument. It follows the same points others have made previously, but adds an important issue: there must be a careful, selective process to create useful requirements, and thus, possibly useful systems. The space, the territory in which this process must take place is narrowly bounded, but better understood:

> "I think the above is correct and is a call to a) narrow the scope of any particular standard to something where agreeing on a common ontology is feasible and b) actually doing the work to develop the ontology for each such standard. In terms of scope, we need to think of 'EHR' or 'health information recording' as something like our scope (i.e. not get deluded that our scope is 'medicine'). Then we (all) need to do a lot more work on the ontology part."
>
> Thomas Beale, 'Subject: Layers of interoperability, OWL and openEHR', technical mailing list, April 2009

Requirements are always moving in EHRs it seems, and for this reason, a process that articulates *and* sources them is needed. If these controversies can be summarised is that requirements are not (complete); they keep moving, they change, their meanings mutate or evolve. There is no answer to the question 'when is a requirement?' Requirements are both a force of expression, pushing their own agenda, asking for their qualities to be observed; and a substance that needs to be guided to take shape. openEHR seems to share this understanding, and it is for this reason that it is seen as a strange, particular case. openEHR is criticised for having taken a longer road to EHRs, but the participants believe it is a more effective one. In question is not the qualities, or the requirements themselves, but precisely the processes that give them these qualities and responsibilities, those that are desired and found useful by collectives.

8.3 Conclusion

This chapter looked at how openEHR engages requirements, and the difficulties of doing so. openEHR shows two principal modes of apprehending requirements: through their stabilisation, or encouraging their emergence. These two modes are complementary, and openEHR shows that for some requirements, it prefers stability, while for the most uncertain ones, a degree of emergence is necessary. The exact degree of stability and emergence of requirements is a political process that openEHR negotiates. Part of this negotiation process has to do with requirements themselves, their body. The substance they hold, or the extent of their expressive force makes them more or less useful to further requirements and openEHR. The interoperability requirement, for example, has a strong assemblage with other actors, though it is not immune to hold multiple meanings for some of these actors. These requirements, then, hold a degree of potentiality depending on the way they are approached, their substance, and their expressive force.

It is the drive towards discovering (and thus creating) the project—even as minute as the work involving seemingly precise tasks (Howison and Crowston 2014)—that joins identities and territories together. If these have been analysed in business contexts (Doganova and Eyquem-Renault 2009), they have not been directly theorised in information systems yet. Some papers, such as Darking and Whitley's (2007), have come close to make the bridge: "the fluid nature of the DBE [Digital Database Ecosystem—an infrastructure-type information system] could create a formative context (Ciborra and Lanzara 1994) which could stifle the ability of the SMEs to engage with and incorporate many of the advanced, distinctive features of the DBE. . . ”. When citing Ciborra and Lanzara (1994), they argue: “once designed and introduced into the organisation, they tend to evolve along paths that are often unexpected and irreversible, subtly changing the ways people design and carry out their work practices, experiment with alternative arrangements and routines, or implement alternative visions and designs. . . .” These visions and designs are not settled or well-defined. Their various degrees of institutionalisation (e.g. in code) pull the project differently and with different force. openEHR recognises this by the way they have decided to engage with requirements: it is a constant work in progress.

The next chapter discusses potentiality of requirements by proposing the concept of virtual requirements.

Part IV

Complexified Requirements

Chapter 9

Discussion

Change and stability, time, space, uncertain territories, political alliances—these are some of the actors that influence (and are influenced by) requirements in openEHR, each to various degrees. In this respect, requirements cannot be defined as clean objects, framed and independent. As the analysis of openEHR shows, they engage naturally with other actors and enter (and alter) assemblages. In this way, they bear a responsibility in shaping realities, moulding the world according to their interests. However, it is impossible to clearly say: “this is a requirement” with consequence. They are identifiable as requirements, but are far from the clean cut, stable, description statements that traditional requirements engineering pictures. Their influence on reality is so meaningful, even capable of shifting a project’s sense of identity, that their force needs to be negotiated, their place contested, their timing put in question. Instead of being static or stabilising, requirements are debated and changing, as if the project was trying to accommodate them within itself.

In this chapter, the temporarily disentangled themes of identity, territories, and modes of engagement are reassembled. These themes are, indeed, closely related, so much so that it is difficult to attribute correlation to (or via) any one in the shaping of requirements (or what they come to do), and thus partly, of reality. Perhaps, the key is to understand requirements as social phenomena; a human attempt to draw the world. If so, how else could it be, but messy and incomplete?

Would this mean that requirements should be rejected by engineering as a lost cause or a false trail? Perhaps requirements can be re-vindicated, re-accommodated as powerful actors in a world now richer with problems and richer with solutions?

This chapter explores these two alternative visions: requirements as deterministic and simple, but complexifying objects, and requirements as complex actors in-becoming. The conclusion arrived at proposes requirements to be virtual and multiple, just as Schrödinger's cat is alive, dead, and paradoxically both at the same time (if only time were an instant), so requirements are both actual and competing virtualities.

I structure the chapter as follows:

Entangled bodies of requirements. A requirement represents an assemblage of various bodies and actors driven by desires. This has an influence on the assemblages' identities, their mode and logic of exploitation, the territories that are contextualised through them, and the collective intentions embedded in a project.

Requirements *in-becoming*. The two principal views are contrasted: traditional engineering sees requirements as determinate or determinable objects; the Deleuzian inspired view that requirements hold indeterminate potentials for shaping realities.

Multiplicity of requirements. Requirements hold as attributes spatio-temporal multiplicities. The multiplicities make possible the competition between potential requirements.

Towards a theory of virtual requirements. The nature of some requirements is distinct from others. This final section offers a new theorisation of requirements and proposes that their nature be considered virtual. The virtuality of requirements acts as a bridge between competing potential becoming and their actual realisations. Virtual requirements have various managerial implications that should be considered and researched further.

9.1 Entangled Bodies

This section ties the model emerging from the analysis of the three principal themes—identity, territories, and modes of engagement—to Deleuze and Guattari's theory. These principal themes act as a generalised assemblage for requirements in openEHR. This section discusses the relations in the assemblage and contrasts them with the assumptions made by traditional requirements engineering.

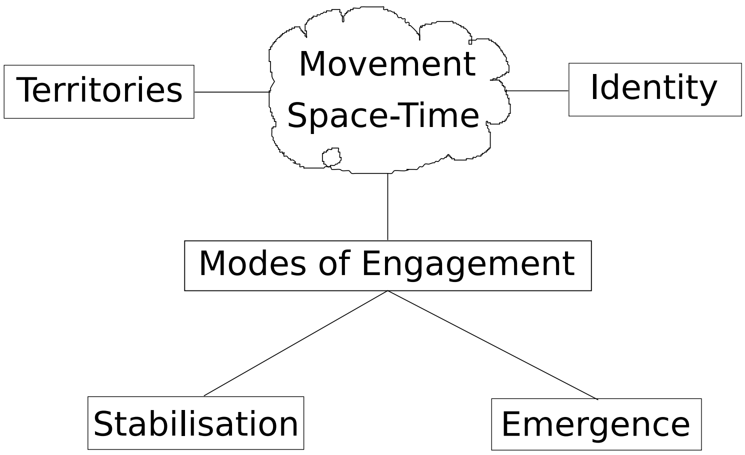

Figure 9.1: Theoretical themes from coding. Being in the world: identity and values, modes of engagement to contextualise the world through shifts and movement

Traditional requirements engineering tends to look for a controllable, consistent set of requirements that confidently describes the system and its relations to the world correctly (Boness et al. 2011). These objects receive a large amount of agency once they are purified through a process of definition and legitimisation, for example, often becoming part of a contractual agreement (Skene et al. 2004). This contractual agreement cements requirements and can give them a deterministic, normative agency over future coding stages and even organisational practices (Boehm 2002). This can be useful for uncomplicated requirements. However, complex, uncertain requirements are inherently problematic because they need to be apprehended, approached and intimated carefully, requiring a certain level of flexibility in their handling (Mellis et al. 2013). The requirements engineering process, if overly constricting, is not well suited for apprehending their nature, their meaning, or their effects. There is often little time to problematise requirements since they must be turned into a solution in a pre-established amount of time (often short), often leading to ways of dealing with 'vague' requirements (Baskerville and Pries-Heje 2004).

openEHR works around the problem of uncertain requirements by explicitly recognising their atemporal nature (in the sense that their nature cannot be defined in a bounded time). openEHR directly incorporates uncertainty in their requirements' substance (the versioning in the CKM) and in their expressive force. For example, the Adverse Reaction requirement needed to stabilise some elements so that they can be further developed depending on the context of use (c.f. section 8).

The way requirements are incorporated depends on two aspects which openEHR negotiates: identity and territories. The first aspect, identity, relates to the project's potential sourcing and articulation of requirements. The influence of identity on development practices and the purpose and portrayal of projects has attracted some research attention (Tyworth 2014), though this research is more in line with performative identities that are not settled (Doganova and Eyquem-Renault 2009; Orlikowski and Levina 2009). The negotiation is as much on-going as openEHR's identity is unstable (in the sense that it reflects often on its goals, purpose, and place in the world). As on-going as the negotiation might be, the influence on how requirements are sourced and articulated is direct: 'what collective is openEHR attracting to source requirements?', 'how much expressive force has the community to articulate requirements in their own way?', etc.

The second aspect, territories, is less about potential (though it does have elements of potentiality). Territories are physical in nature, but their contextualisation, the understanding actors derive from them can also be seen as interactive frames (Davidson 2002; Urquhart 2001). This study, though, went beyond the comprehension and justification by human actors and was more concerned about the spaces in which requirements—as their own actors—can be realised. It is a more pragmatic concern, and many participants are concerned about the level of guidance they needed to articulate requirements in specific territories. The practical side of this aspect is the realisation of openEHR's requirements in relation to the specificities of the strategic (or simply possible) spaces openEHR could settle in. Unlike identity, this aspect may have strong implications for specific clinical requirements: how to implement this or that archetype to this hospital's EHR systems. Other territories have a lesser influence on specific requirements, but a wider influence on potential requirements. For example, 'how can openEHR respond better to the articulation of requirements by national EHR programmes?', or how to localise openEHR in Spanish contexts? The relation this aspect has to requirements is relative to openEHR's ability to contextualise territories, to apprehend them. The contextualisation of territories, thus, involve an understanding of the terrain's realities and of its realisation and evolution under openEHR's influence. Territories, in this sense, are made, transformed, and so evolve.

Time and space have a supportive hand in how requirements are sourced and articulated, and on their larger assemblages (Boehm 2002). A relativistic understanding of time might explain the original disconnect between the intentions of the software engineers and the realities of the actual practices and use of requirements (Akrich 1992). openEHR makes the problematic of time, both explicit and implicit, in its treatment of requirements. It is explicit when it versions its clinical requirements. It is implicit when it avoids design by committee to source and articulate requirements. The implicit problem of realising requirements by committee being that time is forgotten, and requirements are too far away from the 'front-line' to matter in the way it was hoped for originally. The design by committee is an analogy to the game of Chinese whispers, played with a select group of actors who have 'othered' (in the excluding sense) potential actors (Klecun 2008).

Values are also a principal supportive aspect of requirements. For example, values intervene in the sourcing of requirements through their effect in shaping the project's identity and the building of its community. How that happens is often dependent on the values attributed to the kind of requirements that the project participants want to see and the way requirements are given their expressive force. In that sense, values are both used to evaluate what is 'good', and perform competing alternative realities (Thevenot 2002).

The emergent model shows requirements' assemblages (c.f. figure 9.1). The associations these assemblages make are essential to the sourcing and articulation of requirements and for their incorporation into the openEHR body. The emergent model is not deterministic, positivist, constructivist, or phenomenologist. Rather, it is relative to the realisation of requirements at local states (specific territories and their understanding), and global (the influence of openEHR's identity). In this sense, the realisation of the meaning of any requirement is not fundamental, but relative and their meaning is entangled with the rest of their assemblages. The relativity, in turn, allows for the creation of competing realities about how requirements (and their assemblages) should, would, or can become.

In this sense, openEHR and requirements incorporate each other. Requirements need to be assembled into the project, and the project needs to be assembled into the requirements. In other words, what a requirement can represent is not only itself (or a desired state of the world), but the project as well. This incorporation opens the door to a perspective that sees requirements as undefinable (and unresolvable) actors. Undefinable actors are actors whose meaning cannot be held down for a length of time without degrading the meaning of that actor. The degradation of a requirement may be due to its loss of realisation potential in specific territories and contexts, for example. This has two consequences. First, requirements cannot be defined by analogies (e.g. requirements is like. . .). Requirements are as much their processes and the assemblages that realise them and are thus instantiations of a reality (not a reproduction). Second, requirements are necessarily in-becoming; by nature, they cannot be tied down since their assemblages are not completely definable or resolvable; a part may always remain whose potential has not been completely met.

9.2 Requirements *in-becoming*

Two accounts have been made of requirements thus far: traditional engineering, and what has been called until now, for lack of better words, requirements in-becoming. This is a straw man polarisation. The principal difference between these two views is the degree of 'becomingness' that requirements, as actors, are allowed or encouraged to pursue. At first, this might seem like an ontological relativism: the world would appear to have various epistemic-ontological properties of becoming (Thompson 2011). However, following Deleuze, since the world is immanent to itself—thus made from its own self—this could not be. In this sense, there is no subjective-objective entity, since these would necessarily evoke a transcendental (and thus external) 'Being' that could, even for a limited time, make universal (or singular) value-judgements as to what is 'normal' (Smith 2012). A true becoming of a being will only be so if it is in a differential relation to itself and others according to levels of intensity. In other words, there is no fundamental concept of being that can be capably described by analogy or equivalence; there will always be a sense of derivation, not only in substance, but also in expressivity.

Therefore, the argument presented in this thesis adopts the view that, even though all requirements are in-becoming (i.e. from either account), it is the way they evolve that is different. In other words, they are "capable" of more or less potential (Smith 2012, p.41). The epistemological position here then is one which indirectly influences ontology: if all is in-becoming, it is the actors and the statements they can make about the world (which is itself in-becoming) which are different (in intensities). As in openEHR, some will aim to reproduce realities as they are, and others will look to their evolution.

The spiral model, and the flexibility it introduced can be seen as a step closer to acknowledging the becoming nature of the world (Boehm 1986), adding flexibility to evaluate each activity through the life cycle of the software engineering project (Nilsson and Wilson 2012). Agile methodologies attempt to take it a further step, bringing to the fore a discussion of the role of software development methods in the wider organisational setting (Conboy 2009). Agile methods, though, maintain the spiral model's intention of sequentially incrementing understanding towards a unitary conception (Nilsson and Wilson 2012). From the outset of a project, whichever traditional methodology is used, be it waterfall, spiral, or agile, the intention is to build a set of code around a framed, incremental knowledge base representative (and deterministic) of reality. Figure 2.7 shows the unitary accretion of concepts towards the 'golden' representative. This view has many consequences. For traditional requirements engineering, since there is a 'golden' representative to look forward to—the ultimate increment—requirements are conceptually whole objects, assumed as well-described (as a confidence factor) and pure. In contrast, contemplating requirements as in-becoming treats their existence without assuming they are whole, and thus, what they could become. In other words, becoming indicates directionality, not intentionality (Massumi 1992).

The requirement statements which are assumed whole (such as "The system should..."), or their SSM counterpart of the world view description (the *Weltanschauung* (Checkland 1985a)), are bounded within particular time-periods. Change is seen as disruptive, not as a chance to usefully map realities in time. When attention to change is paid, it tends to be defensive and mechanistic; requirements are taken as contractual agreements or are tested recurrently through interactions with the client (Cohn 2004). Defensive change stamps requirements and the technological actor that is supposed to ensue with a 'consume-by' time. The software as socio-technical object is supposed to fit within the intended design assumptions; any deviation is either the result of incorrect assumptions, or a user's misappropriation of its use (Akrich 1992). In other words, before technology is even out in the market, it is already out of date because of the assumptions of use made before its implementation (Ciborra 2002b). Yet, even dated systems persist, resisting change and remaining often leaps and bounds beyond their expected time (Bisbal et al. 1999), despite (or because of) drifting and work-arounds (Ciborra 2002b; Orlikowski 1996; Pollock 2005). In other words, systems accommodate change out of necessity (but not needs), far away from the high-level planning that were supposedly to determine them in the first place.

To survive for such a long time, an information system (as a technological actor) must have connected itself into the organisation (the context of its use) and made itself indispensable.[1] Indispensable in the sense that, once accommodated by the other actors in the organisations and incorporated into their practice, the requirement's forces that underlie such a technological actor, both substantial and expressive, become assimilated by the organisation as part of its becoming as expressed, for example, in phenomena of drifting and work-arounds (Orlikowski 1996). Is this all that requirements are capable of? Could requirements reclaim their ability to become *other*, even as instantiated in the organisational actor? Could requirements be more than deterministic edicts of law, imbuing code with an over-powering and legitimised agency (Lessig 2006)? Deleuzian conceptions of substantive and expressive force provide some answers. As set out in section 3.3, substantive and expressive forces are akin to material and materialisation forces debated in contemporary information systems research (Orlikowski and Scott 2013), though in a particular, Deleuzian flavour (Massumi 1992; Smith 2012).

A requirement, defined as a force based on its substance and expression, invites a certain correlation between its materiality and materialisation. In other words, the correlation asks of the body: "what are the affections of which it is 'capable" (Smith 2012, p.154). A plank of wood is as much a boat as it is a table—though less a motor car; a hammer to hit nails can also be used to dismantle walls, or act as door stop. The potentiality for use is guided by its substance, but not determined by it. So it is with requirements. Their substance guides use (a versioned archetype, a coded prototype, a written specification, or a narrative shopping list will each hold different potential for becoming). A requirement's physical materialisation is more plastic than other materials. In other words, requirements can change, both in the ways they exist in the world (their identity, their substance, their associations), but also in the way they are expressed (how they might become satisfied, how much and how far they are manifested). Coded requirements will tend to materialise differently than their non-coded counterparts. Requirements are, in the end, not

[1] It could be for negative reasons such as the incapacity of the organisation to adapt the assumptions of the software, so that it *must* be used.

subject to their initial purpose, nor are they to their initial expressions; they are translations in-becoming better described as directions rather than the elusive completeness and determinacy of intent.

The materiality and the materialisation of requirements becomes entangled, almost inextricably so. Creating requirements can have only so much influence on the material and materialisation of requirements. The initial deterministic requirements engineering can empower requirements until the process lets go of them and turns them into the 'wild', where they may, or may not relate to the evolving environment as much as they were intended to. Following this route, a requirements process is not there only to define requirements and explore the purpose of a project as contemporary requirements engineering proposes (Letier and Lamsweerde 2004), but also to understand the potential of its requirements by helping shape the substance and expression of the potential they should embody. If the traditional approach to requirements engineering aims essentially to narrow and stabilise (abstraction extracts (Guattari 1995)), open processes consider requirements in their own terms without suppressing them unnecessarily.

The analysis chapters that precede this discussion are each dedicated to identifying specific facets of requirements and showing some of the possible multiplicities they allow and incite. In each of these facets—identity, territories, modes of engagement—requirements are shown to have different natures, roles and purposes. Each facet allows a glimpse of what the project could become. In other words, even if requirements were reduced to any one of these facets, they still would make sense in their own terms. The requirements in each aspect have some specific weights and are able to affect the project in their own interests. One consequence is the conclusion that requirements are independent, as any other actors can be. They possess no human intentionality, but that matters little once control over requirements slips. It is in their evolution—those new associations which give them an evolutionary advantage—that they become free of over-powering intentionality, much like any human being when reaching for adulthood. In this sense, requirements cannot be assumed to be naive, whole objects since no total control over their consequences can be exerted. Parnas and Clements (1986) all but concede this point when they ask for rational processes to be 'faked'.

In any case, would it really be useful to control and rationalise requirements? To tie down their becoming so that informaticians could code their needs or those or their clients in predictable ways? openEHR offers evidence to suggest that this is not necessarily desirable in two ways. First, this becoming is necessary for requirements to pursue their desired body (their full assemblage). Second, without the multiplicity that becoming affords, requirements cannot become or operate as full representatives of all their potential realities; a prior and pre-emptive choice would have to be made at some point to allow for the definition of the requirement (which would always be premature for a requirement's becoming). In other words, to rationalise a requirement is to freeze it, stopping short any future rethinking about its own desires.

Such rationalising may be seen as necessary for future manipulations of a requirement, especially the subsequent activities of coding and implementation (which could be seen as a fake becoming that allows outside intervention to update the requirement). After all, it is simpler to implement something which is rationalised, since its possible range of action is determinate. It may also be easier to accommodate change in a concrete technical actor, rather than a noisy set of requirements. Having entropic, indeterminate requirements would pose problems when explaining what a programme does or should do differently. The question becomes almost an ethical one: what over-powering agency gives the right to end a requirement's life prematurely? There has to be an actor, or assemblage, transcendent beyond, untouchable, which takes upon itself the role of deciding the requirements' substance, and how (and how long) they may legitimately express themselves. For the design-by-committee, it is a democratic process by a select group of experts. Deciding on a requirement is not an innocent decision, since, in its wake, a kind of assemblage favourable to the decider takes shape (the committee becomes re-enforced). The potential for a decision is not wrong, it is part of the responsibility requirements engineering assumes (Urquhart 1999). But this decision is based on the basis of accountability and it may not be right for every project. Thus, openEHR seemed comfortable to dilute the responsibility of decision-taking by delegating it to the community. This is mainly due to the amount of uncertainty of clinical requirements, but also to the uncertainty of what its community needs.

The curious thing is that uncertainty was both in openEHR's core requirement (the language used to describe clinical concepts), and the more peripheral requirements (the clinical concepts themselves). Since its inception, the project was contested, as if its existence was harmful to others. openEHR's requirements have never been free from the project's stance in the world. Perhaps its idea of separating content and meaning carried with it a political affiliation, as if it may taint other projects with an affliction. In other words, the project's requirements were in an ambivalent position: on one hand, they were articulated to solve a problem in the market, and on the other hand, the market itself was recalcitrant to accept that solution, and possibly the extent of the problem. In this sense, requirements were not wholly dependent on openEHR, they also relied on its accreditation as a worthy solution. Who should decide then that openEHR's position is right or wrong? The market? The clients? The project's participants? The requirements engineers (Urquhart 1999)? openEHR's identity is a consequence of its shedding of part of this responsibility and the assuming of another as an integrator of diverging ideas of what *is* openEHR and how it should evolve. It is the existence of openEHR, its place in the world as shaped by requirements that mostly preoccupies its original founders.

Why should the requirements' agency—to shape a project's 'existence'—be worthy of any concern? The answer is perhaps ontological: requirements are in a world *in-becoming*, and thus requirements themselves are *in-becoming*. There is a loss of control, or at best, an illusion of control depending on how much strength is given to faking rationalisations in Parnas and Clements' terms (1986). Such a world has few determinate scripts to follow. Best-practices become hermeneutic adjustments; 1+1 does not necessarily equal 2 any more, the operand being questioned, and with it, the number it creates. A world *in-becoming* has the consequence of trickling down *becoming* to the actors themselves and the statements they can make about the world. Requirements themselves are in becoming, and so are their performative abilities. The immediate consequence is that requirements are creative of the world, and must be sourced and articulated carefully while keeping in mind their changing nature. Open source has the material substance to allow for this kind of performative attitude, but it must be materialised by the projects themselves. If these materialisations fail, open source projects can hold a resemblance to closed source projects. Examples exist in the literature: developers as gate-keepers (Iivari 2009a). In other words, the world might be *in-becoming*, but if the actors suppress its becoming, no benefit can be derived from its understanding, to the detriment of requirements and the accurate portrayal of reality.

9.3 The Entangled Bodies of Virtual Requirements

Code is law, claims Lessig (2006).[2] This is a curious claim since information is usually associated to a multiplicity of meanings and becomings, being highly interpretable and modifiable; could interpreted (and interpretable) information so strongly condition agency (Kallinikos 2009)? Here, a contradiction seems to appear: if information is so volatile, then why are information systems associated with strong resistance to hard objects, such as drifting and workarounds? In terms of requirements, there are two possible answers: requirements have a strong expressive force, or the technologies which embody requirements have a strong substantive force. If requirements are highly expressive, it means their meaning is in line with the organisation's practices and contexts; there is little disconnect between their expression and their use. In the case of a high substantial force, requirements are inflexible; they could change, but that momentum is held back by the material hardness of a (possibly) over-stabilising technology. In this case, requirements are not easily changeable, but an actor's engagement with them is, and so drifting and resistance can occur (Ciborra 2002b; Pollock 2005).

[2] It is not the 'cemented' code and its inflexibility that worries Lessig, but the pre-emptive decisions about the potential use of a technology.

The extension of the notion of a requirement into associated assemblages of bodies, forces, proxies, and allies makes it difficult to define requirements as framed objects (Barrett and Oborn 2010). The various forces requirements can get involved with are visible in time, but are difficult to conceptualise together. However, the assemblages that requirements are involved are indicative of their behaviour. They might not be framed objects, yet they hold a recognisable identity, a being that cannot be readily defined by its substance, but which exists nonetheless as 'this requirement' because of the affects it is capable of mobilising.

Perhaps, this is the difficulty: sometimes requirements seem ethereal, immaterial beings, too easily controllable, evident even. At others times, requirements are dominating and hard to evolve (Lessig's (2006) code (as a material of requirements), for example). The assemblages have a part to play: entangled though they might be with a variety of actors, in hard technologies such as legacy systems they seem non-pliable and irremovable (Ciborra 2002b). In those cases, requirements seem more 'imbricated' (in the sense of interlocking stratifications) than 'entangled' (Leonardi 2013). Could open source requirements become such hard technologies, coercing people to their dictates and reproducing categories which can end up excluding actors (Deleuze and Guattari 1987)? Possibly, but it is less likely. The analysis around the argumentation of requirements, even basic, foundational ones, seems to indicate an embraced heterogeneity, catered for by the openness of the code which allows evasions into new becomings (the ability to fork project's requirements, for example).

A critique of this complex, associative view of requirements is that their bodies are imagined, that they are immaterial. It is true that a 500 page document describing requirements might look more graspable; that may well be the original problem: the weight of requirements might be too heavy to change (Beck 1999). By the time requirements are written down, requirements have changed. Agile methodologies are no different, only they cope better because they require a constant refocus from clients. Requirements, even in Agile methods are still framed in the project in a sequential manner, though through repeatable interactions with the clients

(Nilsson and Wilson 2012). In contrast, openEHR has shown a multitude of ways requirements are given material bodies, only not as a single, referential prescriptive statement. The early framing is discussable, and discussed; the project is allowed to evolve beyond its initial intentions, beyond the initial point of reference, by the participation of any actor with an ability to source and articulate requirements.

Pollock touches on this when building on a citation from Quattrone arguing that simple representations have accrued performative power by revealing complicated aspects (Pollock 2012). They provide an effective 'handle' to grasp difficult, tacit concepts. Quattrone's words suit requirements particularly well: "Graphical representations... are always so partial and simplified that they essentially contain very little; they have little truth in them; for if it ever existed, it has been lost in the process... (Quattrone 2009)". The problem arises when requirement descriptions are not relevant any more, disconnected, and at the same time provide such a strong, substantial and compelling force that overpowers what could be different, perhaps, more useful representations. It is in the body, then, that traditional requirements become mired down by stability: by purifying requirements, they end up holding a strong but possibly inaccurate influence in the project's ability to describe the world. They become negative performative objects and hard to evolve, being 'imbricated'—instead of entangled—in a hierarchy which they come to dominate (or significantly constrain) by their subsequent implementation through coding (Leonardi 2013). A body with organs as an already constituted "organism", and nothing else (Deleuze and Guattari 1987, p.175).

openEHR, in contrast, does not see its requirements as being fully described. Archetypes are versioned, translated, copied and adapted to various national levels. When a requirement is identified, its identity is curated as librarians would. The multitude is accepted: there are various places to talk about requirements and various ways to do it. Even requirements themselves are not tied to a singular meaning of what archetypes should mean, but encompass larger goals which enter into intricate relations with requirements unfolding through discussions. These discussions remain behind in wikis, mailing lists, the CKM, and other projects on source forges that cater to specific aspects of openEHR, sprouting from its trunk.[3]

The requirement's body, then, has a performative action through its negotiated meaning: its representation and its meaning (and their differentiations) add something to realities. In traditional requirements engineering, the representation is meaning *and* reality. Such description of a requirement is, by essence, transcendent: it is both a description of and the essence of the requirement since it is presented as Truth. Evaluation is then dependent on its closeness to a reality; the other realities, and their values, have been left aside (Callon, Lascoumes, et al. 2011; Thevenot 2002). They are cast aside in the process of constructing tables and graphs objectifying the optimal solutions which should describe reality (Boehm 1998; Fraser et al. 2009). In this case, a requirement and its assemblage is valued in relation to the optimal One; reality is ontologically relative only to a higher order which is inimitable but is foundational: the purity of the optimal requirement itself.

In contrast, the agency of requirements in openEHR tends to be messy because of their associative sourcing and articulation. Because of their becoming, requirements are not easily objectified. This is in contrast to Latour's explanations of a phenomenon's objectification, abstracting its most important details and removing the technology's own construction from the objectified phenomenon (Latour 2005). The measurement becomes standard and enforces its take on realities. For openEHR, if requirements were taken as measurements of realities, then these realities would be questionable since requirements tend not to be fixed. Those that become fixed operate a reduction in the requirement's sourcing (less participants are needed for a more or less stable requirement), but an increment in the requirement's expressivity. The requirement, when becoming actualised, also provides potential mappings of the territories and contexts. In other words, a requirement's potential exists inside its bodies.

[3] Source forges are portals giving access to a project's source code and the ability to flag bugs and discuss requirements.

9.4 Multiplicity of Virtual Requirements

If a requirements' body may be in multiple places at once, it follows that its identity and expressive force would also be multiple. To remain the same, a requirement would need to update all its bodies at once and maintain the same forces and their potential in all these bodies; otherwise how could it be the same? Two notions are implicit in this question of multiplicity: one is space (the multiple material bodies); the other is time (the time it takes to update changes). Space, together with time, allows for the multiplicity of requirements. This means that through time, the identity of requirements can be manipulated; time is thus not an external, transcendent variable. The importance time has to requirements is in relation to how vulnerable they are to timely degradation. In other words, *when* can a requirement be said to *be*? The degradation of time is one of the main causes for the disconnection of requirements between their intention (and even direction) of use and their actual use (Ciborra 2002b). The consequence of an untimely requirement would be an incorrect and inflexible portrayal of reality. In that sense, requirements are an epistemological actor in a world in becoming.

To provide an answer to time's important role, requirements should be placed in the wider context of their assemblages. Requirements are not alone to shape the world; other actors in their assemblage can—and should—be used to accommodate requirements. The converse is also true: a project would be better served if it were accommodating to its requirements. For a requirement to be affective and expressive, it needs to be considered in its wider assemblage.

Figure 9.1 shows the emergent model for requirements developed through the analysis chapters. It shows requirements having a constitutive role in the definition of identity and territory. The articulation of requirements are usually done through two modes of engagement: stabilisation and/or emergence. They come together into this strangely shaped cloud called movement/time. Because of their centrality, requirements are present in all these themes: they expand and contract territories to explore, shape a project's identity which serves to source and articulate requirements, and requirements themselves are engaged with and used to engage realities in different ways. But in the middle, time and space take a mediating position, actualised by movements. The more movement is encouraged, the more potential there is for the world to become and the actors evolving in it. The less movement, the slower the actualisation, and the more likely for the world to be reproduced.

For the world to have less movement and more reproduction also means having less multiplicity. The less likely multiplicity is to exist, the more likely single realities are created. In time-induced movements, there is both a moment and a place. The moment, the event is not lost to a locality: there is a pre-existing form, a pre-supposition which is a necessary cocausality in the creation of reality. To be real, there needs to be an interpretation to the space that delimits reality—that is what makes reality multiple. It is through interpretation that sense-making, and sometimes, surprising articulations of a problem and solutions are made, but *made nonetheless*. Spaces are produced (Barad 2013), echoing Guattari's call to look at the production of subjectivities (Guattari 1995). When spaces are produced, just as requirements are produced, middles arise. A middle is a new land—fertile perhaps—but which has no observable limit, no relief to impose a reference to the map. The limits may well be there (e.g. the project is unfeasible, this requirement is too expensive to build, etc.) but they are not known *yet*. A requirement might not be too expensive; part of this knowledge becomes relevant through the observation of realities. The requirement is entangled in the rest of the assemblages, and just like Schrödinger's cat, it is dead, alive, and both.

Time is essential for the production and manipulation of those spaces and the ensuing multiplicity. It is present in many of Weick's studies on sense-making as a locality of time, or temporality of space, which contract the past and future into the present (Weick 1998). The contraction of time creates an ongoing present (Schultz and Hernes 2013). A contracted time manipulates the form and expressiveness of the actors and their assemblages. For example, Weicks' depiction of forest fires involves a rapid decision making by one member of the fire squad, showing that it is not a simple reaction to events, but a temporal interpretation of ongoing conditions that led the fireman toward his unexpected course of action (Weick 1996). There was no automatic reaction, no reproduction of past knowledge, only a new interpretation of an historical experience. Just like Proust's *madeleine* which places the narrator in a nostalgic mood when eating it, actualising (and thus producing) his own memories (and Swann's story) to his present context, the fireman evolved his own solution to the predicament he faced.

Requirements, then, represent contractions of time. They hold a productive process in themselves, just like Proust's *madeleine*. In traditional requirements engineering, the time allocated to requirements is external: the typical case is a client asking for a project to be developed in a limited time-frame. There is little contracting of time possible; the project starts with a finishing time. In open source, this is not necessarily the case. Because of the open license—and if the project's governance allows for this—a multitude of contractions can take place, depending on the number of actors. In other words, the identity of the requirements produced and their project-wide influence is not controllable or restrictive for unexpected becomings. By being close to identities, requirements and their assemblages have strong attachments.

The relation time has with requirements is important to the sourcing and articulation of requirements. Time puts requirements close to the project's identifying questions. The *raison d'être* becomes co-causal to the way requirements evolve. In traditional requirements engineering, the externalised time impeaches requirements' possibility to evolve on their own. Instead, they must hold an early intention and be purified from multiple realities which might complicate the following engineering activities of coding. In the traditional setting, the evolution of requirements tends to be only simulated in the analysts' laboratory, under an analyst's intentional, probing questions. This simulation remains a limited foray into the future and the past; the contraction of time is assumed to be stable and often over-reliant on the past (the system-as-is that is identified with the problems that were the rational for a new system). Time, in this case, is external. In openEHR, this contraction of time can take place.[4] Requirements are evolved in a continuous present; there is no necessary deadline for a project's delivery. In open source, there is no need for such 'delivery'. If the governance permits it, requirements can be, so to speak, their own man. Such requirements would hold two multiplicities: becoming in time and becoming in space. The exhibited multiplicity would make them virtual and becoming performative through the actualisation of that virtuality, indescribable until observed.

9.5 Towards a Theory of Virtual Requirements

This section argues that requirements can be seen as virtual beings in formation. Virtual requirements are physically distanced objects: they are distant from their actualisation and they are distanced materially from the other actors in their assemblages. In this sense, the virtual is not opposed to the real as is often proposed (Aaltonen and Tempini 2014), but to the actual with which it coexists (Zourabichvili 2012). The consequence is that "virtual coexistence must be fully real since it conditions the affect" (Zourabichvili 2012, p.121). The virtual, then, is not detached from realities, but performs them in association with the actual. The actual itself "does not disappear in favour of the virtual, for such a situation would be unlivable [*there would be an absence of time*]; it has, however, become unlocalisable [*there is no specific, determinable space any more*]" (Zourabichvili 2012, p.122). There is a multiplicity of events taking place between the actual and the virtual, multiple contractions of time which "move from conditions to the conditioned, [. . .] from a problem to its solution or, what amounts to the same thing, from the virtual to the actual" (Smith 2012, p.250). In other words, the existential concern of anything is placed on its movements, indeterminate as these movements may be, between the actualised and the virtual. The becoming, thus, is as much a promise of actualising potential as it is a sacrifice of unactualisable potential.

Deleuze and Guattari often use virtuality to understand the realm of potentiality (Massumi 1992). In other words, the graspable aspects of a world in-becoming are

[4] Which does not mean that it always does, only that it can.

the dimensions of virtual potentialities-in-becoming. In that sense, the virtual is an existential part of causality. The virtual exists, and which virtuality is actualised must be done so through the expense of energy by the forces in play. This may sound cryptic, but the main point is that any being is such through movement, just as Bateson argued that pole vaulters appear stable on the pole, but are actually moving in place, becoming still so to speak (Farjoun 2010). That something may be virtual, then, is a promise to become; it is *a* reality.

Virtuality then is not against reality. Ontologically, virtuality exists, just as the Future does to the Past because of the Present acting out. In Dickens' 'A Christmas Carol', when the ghost of Christmas times visits, it is the virtuality of the actors that play out on the real. The story may, if pushed a little, be seen as reality becoming ancillary to virtuality. Where would change be if the Pasts, Futures and Presents did not act out? The protagonist would remain much the same. It is this aspect of virtuality (as manglish as mangle can become) that is useful in understanding the virtual potentiality of requirements.

The virtual potentiality of requirements comes from the realisation that their nature is unsettled. A virtual requirement is one which has a lot of becoming potential, either because it is not entirely described, or because it can rupture and become different; it is not yet enacted. It could, for example, get into new contexts that change the meaning of a work practice. In openEHR, the interoperability requirement was continuously discussed in relation to its reach, leading to questions about the extent of its prominence. In other words, such a requirement is difficult to reach (if ever), but it does pull the way openEHR thinks about the world and its other requirements despite only being partially articulated. Although it is impossible to understand what it means to be interoperable, it is put in practice in the definition of archetypes (e.g. the granularity of their definition and the contexts they can be applied to); the architecture (the archetype definition language); and even the processes (e.g. is the design by committee the best way to reach interoperability?).

For both Deleuze and Guattari, the over-powering forces is their explanation for the existence of State oppression.[5] States—and any hierarchical institutional actor—vie for this ability to reproduce itself on its subjects/descendants. Bourdieu calls this *habitus* (Bourdieu 1992); Wenger, *community of practice* (Wenger and Snyder 2000); and Deleuze and Guattari, *Identity*. In those 'modes', there is a tendency to reproduce the objects the mode produces, and the mode itself. There is little evasion, little virtuality to become other than that which should be reproduced. In a mode centered on reproducing reality, reality is *almost* completely actualised: reality depends on the non-existence of virtuality to remain the same reality.

If this is excellent for non-complex, known requirements, it is prejudicial to virtual requirements. While the non-complex and known requirements would benefit through the repetition of already existing solutions, definitions and best practices, virtual requirements could end up smothered with ill-fitting tools, definitions of meanings, and limiting restrictions. Constitutively speaking, reproducible requirements are coherently different to virtual requirements; they follow another logic and rely on ontological and epistemological simplifications which give the appearance of distinct realities (e.g. the existence of stability versus change as ontological constituent). The immediate result is the assumption that all requirements are born equal, under the same conditions, necessitate the same work, the same methods, follow the same trodden paths, and remain tameable objects. Traditional requirements engineering works well with reproducible requirements, but not so well with virtual ones. The virtual requirement has other desires than the reproducible requirement; it needs special attention to become usefully actualised. openEHR's use of open source seems to cater to some of these desires.

Figures 9.3 and 9.4 caricature the different takes on virtuality that distinguishes traditional requirements engineering and openEHR. Traditional requirements engineering tends to narrow the problem domain early on. In this way, it treats requirements as solutions: their purpose is mainly in providing a brief under-

[5] A modern take on this concept would be the states' intermission on, say, the unilateral exploitation of gas through fracking. Another example is the nuclear waste disposal debate in France and the affected parties, and the various positions the state could take (see Callon (2011)). The State as government is not necessarily the target, but instead an institution in general which positions itself as the universal guarantor of moral right and reason.

standing of the necessary work that later stages need to know about. openEHR, in contrast, had also made early some paradigmatic assumptions, but left open the ways in which these objects could be manipulated at a later time. The openness of openEHR's requirements allowed anyone to evaluate their own needs in light of the project's potential for change.

A virtual requirement is not a solution, it is a problem in-becoming. Virtual requirements have attributes that make them respond in different ways to phenomena. The virtual requirement is not actualised, and by nature, is hardly completely ever so. This is, perhaps, why requirements are so prominent in domains they are usually disassociated with such as identity and territories. The uncertainty of the project—its identity, its context, its use—becomes reflected in the uncertainty of the requirements, and vice versa. This is why in section 9.3, the physicality of the requirements' bodies are so dispersed. Requirements take various, possibly competing, material shapes, but these shapes only look like shapes: none can be said to be *the* requirement (c.f. figure 9.2). This is because movement is interior to requirements; their substance is indissociable to their movement for coherently comprehending them. As Deleuze says: "Leibniz specifically claims for substance *a unity that can be interior to movement, or a unity of change that can be active*. . . [otherwise] we apprehend only an accomplished movement" (Deleuze 2006, p.80).[6] In other words, a requirement in-movement or a requirement in-becoming, are partial to the existence of the requirement's substance. The tangible actor can, at least, only be partially defined virtually, because virtuality is partial to itself (virtuality cannot define virtuality) (Deleuze 2004). The substance can, thus, partially limit the virtual's influence. The result for requirements is that their 'existence' is in flux; there are no clear boundaries with which to define them, no outside and inside. The immediate implication is that, when it comes to sourcing territories, no artificial limit is fair to virtual requirements.

This substantial difficulty—the virtual requirement still depends on the actualised requirement—should provoke a balanced production of virtual requirements.

[6]Unity here should not be understood as completive, but as associative.

That is, some care should be taken not to actualise requirements too soon, or too restrictively for the requirement to be de-virtualised of its becoming. Is the substance always the enemy of the virtual? Not necessarily, and in some cases, it could even be the contrary. This is why, in figure 9.1, both modes are seen to influence requirements at any one time. openEHR, for example, has created a core set of requirements that, even if they evolve, do not do so at the same speed as more peripheral requirements.[7] Yet, this apparent stability is partial to the movement of its archetypes, to the multiplicities requirements can evolve into, and the meanings they can come to signify. In openEHR, the requirement forms part of a movement of continuous integration. The requirement is processual, its meaning and its bodies are entangled and in-becoming. A requirement in open source may be dated, but this does not mean it is *the* requirement. It can be questioned, evaluated, evolved in different contexts. The substance is consequently changed, updated, along with its partial virtuality.

Virtual requirements associate with virtual assemblages. It is not clear with which assemblage or where requirements will be usefully actualised. In other words, an actualisation of these requirements and their associations must be observed to know what the innovation will be; if at all. In this sense, it shares the unexpectedness common to open innovation, user innovation, and bricolage (Ciborra 2002b; Hippel von 2007). The consequence is that the value and the evaluation of requirements are intangible most of the time. Virtual requirements are thus difficult to reproduce since few of the people concerned know the value of the innovation (Ciborra 2002b). Virtual requirements, because they are distanced to their actualisation, tend to be held as outsiders and minoritarians. This minoritarian logic makes virtuality disturbing (and not necessarily disrupting). Virtual requirements are *othered* to their realisation, seemingly ill-fitting, before becoming actualised and possibly mainstream, reproducing themselves and their views of the world.

[7] Precisely, the Archetype Definition Language and the paradigmatic conception of what an archetype is are slower moving core requirements, while archetypes themselves are faster moving.

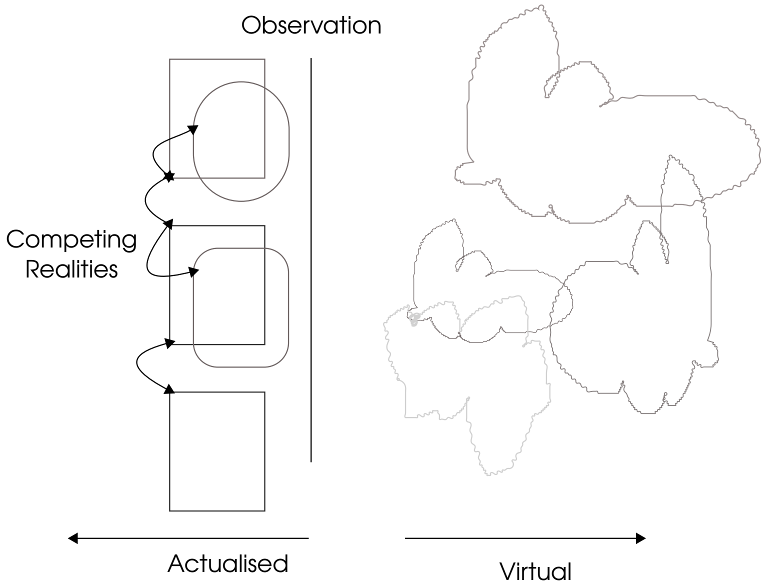

Figure 9.2: Virtuality of requirements

9.6 Conclusion

This chapter presented a theory of virtual requirements. The first step was to theorise, away from the case study, the nature of such requirements. In doing so, a principal assumption, tied to theory, was made: that the world is ontologically in-becoming. It is a processual understanding of the world which sees requirements as having a performative influence. This performative influence of virtual requirements is dependent on various factors such as the multiplicity and potentiality. The virtuality of requirements acts as a bridge between potential becomings and actualisations, that if carefully considered, can be used beneficially to reduce the disconnect between the intended use of requirements and their actual use.

	Deterministic	*in-becoming*
Ontology	Not shapeable but findable. Almost purely formed at moment of 'capture'	Evolving nature; its own self, but in assemblage of multiple actors.
Epistemology	Affords requirements a contradictory nature: tamed and powerful, true descriptor. A requirement's efforts are based on maintaining its own purity of substance and expression. Causality is evident: the requirement's own	Requirements as performative actors in an evolving world. Need support from an assemblage to follow desires. Causality is rarely evident and flows through several dimensions. Active participant in its assemblages development. For example, asks 'How should the project look like?'
Agency	Contradictory and assumed. Pre-development: no agency, lifeless, overpowered. Post-development: true descriptors, complete (assumed) agency over space (program) and time (schedule)	Complex agency. Degrees of virtuality. Possibly useful early on, needs participation to get potential substance. Requirements can be sourced and expressed to be more or less forceful. Requirements can be made stronger, more flexible, more hospitable. An evolution in progress
Spatial connectivity	Framing done early on	Space enlarged or reduced in motion. Expansiveness of territory related to the evolving nature of requirements
Temporal connectivity	Permanence of identity. Energy spent on maintaining purity of design. Temporality subjugated to rooting of requirement, leading to reinforcement of identity in time	Contraction of time but differentiation of nature. Time is local to performativity of world, i.e. it is not an external variable of constitution of being
Essence	Contained in the statement	Evasive and virtual through multiplicity

Table 9.1: Operationalisation of requirements' nature through deterministic and *in-becoming* theories

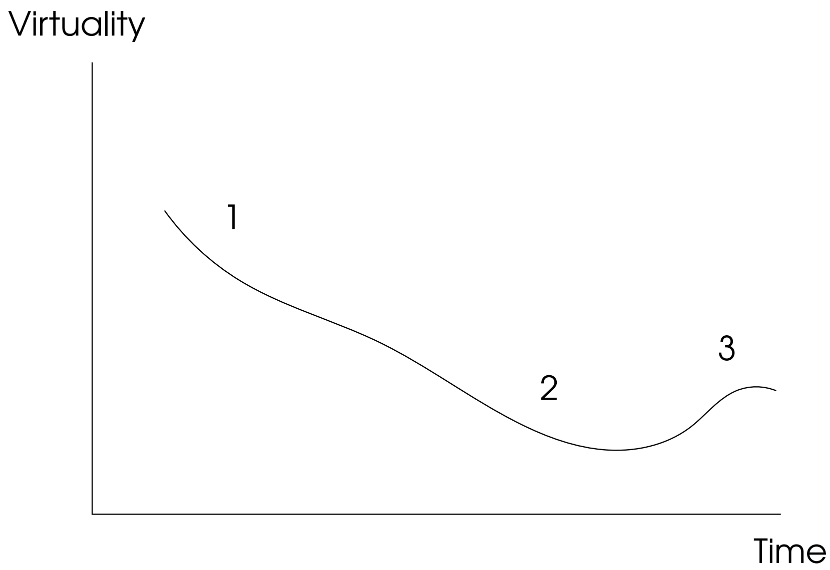

Figure 9.3: Potential for becoming of requirements in traditional requirements engineering projects: 1. scope is narrowed at onset; 2. requirements are implemented; 3. drifting, work-arounds, accommodation

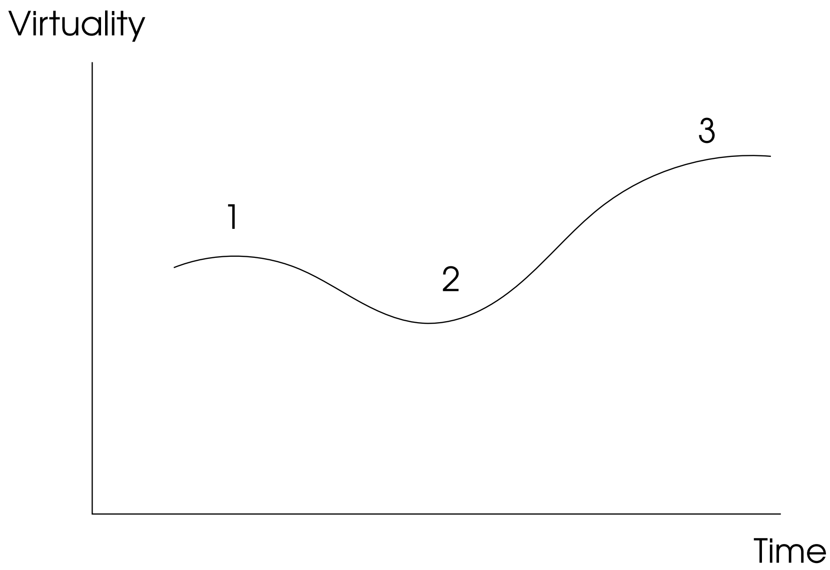

Figure 9.4: Potential for becoming of requirements in openEHR: 1. scope is narrowed along general project identity lines; 2. core requirements are stabilised; 3. usages are revealed

Chapter 10

Conclusion

This chapter wraps the thesis. The original motivation was to find an alternative to the prevalent, traditional understanding of the nature of requirements. The problem with the traditional perspective is that it does not take into account the contemporary evolution that information systems are undergoing and the evolving organisational complexities. Assumptions of stability which underlie established notions of requirements engineering are useful because they streamline difficult processes, but run the risk of over-simplifying reality and disconnecting the actual use from the intended use.

This thesis has searched for another way of understanding requirements, one that would not be based on assumptions of stability. In contrast, this research has looked to the notion of change for a reconceptualisation of requirements; of what they are, and what roles they can play. Change, as conceptualised by the notion of becoming from Deleuze and Guattari, does not believe in the prevalence of stability. In a world in-becoming, there is no normal state for things to recur to. There is, therefore, no pre-defined judgement on what a normal requirement would be or act like.

This is where, apart from a challenge to mainstream notions of requirements, this study hopes to contribute. To propose a different nature to requirements has consequences. A different nature would necessarily imply a reconsideration of the management of requirements and their processes. Also, their place within (and without) organisations and software engineering would have to be reconsidered. The hope is that a better understanding of requirements would better trace realities for information systems development.

Although this research has taken a specific domain of enquiry—open source health IT—the theoretical framework used can potentially generalise its findings to other domains. In this sense, requirements are not so special, not so transcendent that findings about their nature might not travel beyond their own-selves.

This chapter is structured as follows:

Limitations. This study suffers from some limitations which need to be discussed.

Contributions. This section lists the contributions to information systems, open source, software engineering, and, more immediately, to practitioners.

Further research. Presents some ideas for further research.

Summary. This section summarises the thesis.

10.1 Limitations

This study suffers from a number of limitations that must be considered in the context of the thesis.

- Further work on differentiating Deleuze and Guattari and ANT could be revealing. This thesis, though based on concepts of becoming, mutliplicity and virtuality, is also greatly inspired by ANT and STS studies in general. It is not in the scope of this research to determine the diverging points and how compatible the two are.

- There is a link that is implied between requirements and innovation. It is not made explicit because it would complicate further an already complex topic. However, the link is present, especially when the analysis and discussion cover the internalising of requirements. The innovation literature would add interesting aspects to this research, but they would also involve new actors—economic ones particularly—that could be confusing.

- openEHR is a project where uncertainty is high, and virtual requirements are particularly useful to uncertain environments. In this sense, this research is specific to such projects which deal with unknown environments.

- Discussing the importance of uncertainty to virtual requirements questions the proportion of IT projects which cater to uncertain needs. There is a subtle trend, worrying for some, that sees the evolution of information systems into 'tethered appliances' instead of the generative platforms they have been (Zittrain 2009). The surge of ubiquitous software has also brought with it a tendency to close down platforms. Many of these are becoming marketplaces—'app' stores—on which it is almost impossible to install or experiment on 3rd party software. The problem is not the closing of the platform so much as the difficulty to experiment and propose information systems that do not conform to strict guidelines. This has a direct consequence on requirements because they have to conform to these guidelines. Some browsers, for example, have been banned from certain 'app stores' for representing a competing solution to the platform's official application. It is unclear how open source—often based on *bricolage* and generative processes (Ciborra 2002b; Zittrain 2009)—will cope with this.

10.2 Contributions

Because of the multi-disciplinary aspect of this research, the contributions range several fields.

10.2.1 To Information Systems

- The principal contribution to information systems is the ontological study of requirements as actors in their own right. This thesis associates requirements with two important middle positions. One 'middle' position sits requirements between the actual socio-technical worlds and the potential socio-technical worlds. In that sense, they are socio-technical objects (Akrich 1992). Requirements mediate mappings of realities and a materialisation of these realities. They represent an evolution of realities, and the means to evolve them. The other middle position is within organisations. Requirements and their assemblages entangle themselves deeply to the conception of their selves and their possibilities with realities (Doganova and Eyquem-Renault 2009). Requirements can be intentional (strong enforcement of their selection), and directional (possible guidance, but freedom to become).

These two middle positions show the difficulty for anyone to understand or encapsulate requirements, and therefore participate to the redefinition of the concept of expert within software engineering (Callon, Lascoumes, et al. 2011).

- Treating requirements as actors instead of objects places them in the middle of diverse assemblages and mechanisms to which they participate. Such participation is more an entanglement than an imbrication in the sense that it is not a cleanly describable relation (Leonardi et al. 2013). Hernes (2014, p.44) reflects on this matter by making processes constitutive of the world. The analysis and discussion depict requirements as entangled with disparate actors, such as the identity of a project and its own relation to the world. The direct implication to research is that actors under study need to be looked at from a temporal perspective as if the past and the future were dimensions of the present (Deleuze 2004). What this means is that entities possess a vitalistic capacity to act on their present by considering both their past and future. In other words, the two dimensions are enacted in the present. For example, the virtuality of a requirement is not only what it is in the present or what it might be in the future, but how the future (and past) participate in the creation of the present. For example, the interoperability requirement affects the sourcing and articulation (in the present) of other requirements because of the possible ties it might create (in the future) between unknown actors, which is one of the reasons why some people in the 'front-line' were vexed by the lack of individualisation of their own requirements.

- This entanglement shows the complexity of processes. In effect, it poses the question of the difference between entities and processes. Requirements might be an analytical 'thing' that can be discussed and described, but only partially so precisely because it acts like a process (and thus is performative). Pickering's (Pickering 2011; Pickering and Guzik 2009) theorisation of the mangle of practice shows the productive preoccupation that actors have. The results of the thesis tie well with STS studies, especially Callon's et al. (2007) work where the value (and therefore the identity) of things is discussed and performed.

  Thinking over the value of requirements opens the door to their politicisation. Despite being often considered as technical and scientific objects, their evaluation is often tinged by politics (Law 2002). While this, in itself, is not novel (e.g. (Markus 1983)), the way politics act on requirements or hold out from acting is. Previous work had framed requirements looking at quick and up-turning shifts in thinking (Davidson 2002), often from the participants' point of views and with clearly, though discussed, roles (Urquhart 1999, 2001), or as objects that intermediated between different actors (Barrett and Oborn 2010). Instead, openEHR showed a, seemingly, never-ending learning experience in which settled discussions reappeared continuously (for example, on its identity). There is, thus, a long duration element to the way actors are understood and the role they take in enacting realities.

- The literature review is a first step to build a foundation onto which requirements can be theoretically discussed (Jarke et al. 2011). Further work in this area could extend the review to take into account other design theories such as Mathiassen and Sorensen's (2008) view of information as services and contrast it with Kallinikos's (2011) concept of functional simplification. Indeed, the analysis and discussion point towards a complexification of requirements where the cut-off point of what *is* the requirements is postponed both to more concrete situations (instantiation), layered (archetypes as clinical meaning), and where the processes try not to funnel or close requirements (archetypes can be opened and discussed even after reviews). This is shown by both the oncological and allergic archetypes discussed by participants (c.f. section 8.1), where involvement was opened to unexpected actors and the situation of application discussed.

- This study contributes to an operationalisation in information systems of Deleuze and Guattarian concepts. This aspect would warrant a further discussion to how they relate to existing aspects of processes (e.g. Weick's (1996; 1998) learning or improvisation), to the matter of Being's opposition to bodies when discussing entities or actors.[1] The view of requirements as an assemblage questions the prevalence or the influence of dominating structures that might be more challenged than usually thought. Orlikowski and Scott's (2013) study is revealing when it considers the competition between different value systems in a vein that is not unlike that of Boltanski and Thévenot's (2006) work. In this thesis, the two systems were the 'modes of engagement' that at times competed, and at others, cooperated. Because of this competition, there was no idealised system or development methodology, thus contributing to Ciborra's call to look for the 'drifting' and the messiness of the world (Ciborra 2002a). [2] In this sense, openEHR was not a single project with clear boundaries from the beginning, but evolved, echoing Simone de Beauvoir's (1976, p.13) words: one is not born a woman, one becomes it.

[1]Deleuze and Guattari repeatedly criticise the use of Being (*être*, *il est* in French—'it is'), instead preferring the term of bodies or the conjunctive (and thus also disjunctive) 'and' (*et* in French). The reason they cite for their preference is the lack of immanence in the concept of Being because it describes a well-formed 'whole' that transcends the relative between itself and other, 'external' things (Deleuze and Guattari 1991). The result is that Being can be supposed; it pre-exists and dominates not only its own form and expression, but also that of others. Such well-formed Being is already a category onto itself, which grouping of things cannot be deconstructed. Bodies, instead, can present this amalgamation of functions that are not necessarily hierarchical—referred by Deleuze and Guattari as the BwO (Body without Organs) (Deleuze and Guattari 1987).

[2]Even the use of sourcing and articulation to describe requirements tasks were expressly left somewhat interdependent.

10.2.2 To Open Source

- The principal contribution to open source is a depiction of the negotiated meaning of what it means to 'do' open source. Rather than settled, open source is experienced and discussed (Coleman 2012). In openEHR, part of this discussion showed that the project challenged the role of the expert, and particularly, the processes that recreate it (Feyerabend 2010). Thus, the link between the user and developer (of meaning or code) is both blurred and re-enforced, leading to a co-creation of information systems (Teege 2000). To do this, openEHR repeatedly asked itself about the adequacy of its governance model and its support to participants and users alike. The consequence is that open source is not so much a method or a context than a minoritarian approach to question the establishment and derange the status-quo (c.f. section 6.3). This work evidences how open source can contribute to disrupt established players in a field.

- Co-creation in openEHR was not as 'gated' as previous studies in open source indicated (Iivari 2009a; Laurent and Cleland-Huang 2010). This is, perhaps, because openEHR took some care to widen participation as much as it could or knew how to try (although this too was debated, sometimes heatedly). For example, section 6.2 showed there were multiple shifts between cores and peripheries (there is a questioned centrality to the project). This influenced the sourcing of requirements greatly by altering the prominence of actors. Even the founders were not exempt of criticism (and neither did they try to smother the criticisms), questioning the possibility to establish strong, authoritarian figures in open source, such as the benevolent dictator for life (De Laat 2014).

10.2.3 To Software Engineering

- Not treating requirements as tamed objects puts into question the philosophical foundations behind requirements engineering. The purpose of requirements' processes can be rethought; the objective is not to identify requirements through one-off processes (whether waterfall, spiral or agile), but to circle around multiple potential realities. In this sense, new processes (possibly influenced by open source practices) could take flexibility into consideration early on. Thus, requirements engineering would not be responsible for delimiting the project's space at the outset, but identifying the overall direction and putting off boundary-making decisions to a later, more beneficial time.

- The implication of the previous point contributes to a change of focus on the activities that take place around requirements. For example, instead of being concerned with signing off requirements, activities would focus on the sustainability of the project to gather and incorporate requirements. In this sense, a more global perspective on the influence of requirements would be necessary. Such requirements would not be external objects with well-bounded influence since their sourcing and articulation would also affect the project's governance and processes. Further research in the line of Ostrom's (2010) institutional studies focusing on sustainability could be helpful to rethink successful software engineering projects (Curto-Millet 2013).

10.2.4 To Practitioners

- One of the main points of this thesis is that management practices should consider requirements as full actors. Part of the reason they can be full actors is their capacity to form unexpected assemblages into potential realities. The question is the following: how should an increase in applicable contexts for information systems be handled respective of requirements? Ciborra makes a convincing argument that the success of information systems has happened despite its management (Ciborra 2002b). Zuboff (1989) said much the same years before when she showed the evolution of management principles and the various engagement they inspired among workers displaced by technology. This does not mean a *laissez-faire* management, but one that internalises external actors and requirements into the project along some practice guidelines and governance structure. This is where the role of expert becomes democratised (Callon, Lascoumes, et al. 2011), and where *bricolage* takes its full potential (Ciborra 2002b).

- Becoming can involve fragmentation which is a possibility particular to open source. This can be positive (a search for potentially more fruitful routes by side projects), or destructive (emptied bodies instead of bodies without organs) (Deleuze and Guattari 1987; Deleuze and Guattari 1991). This is why there must be a cautious search for heterogeneous requirements, one which does not invalidate the search for actualisable potentiality. The point of virtuality is to go beyond boundaries where a new expression can be born. Such a search has its dangers: away from references, it is easy to lose the point and collapse into the old references, or building harmful ones.

10.3 Further Research

- It would be interesting to follow openEHR's implementation in concrete settings. The redefinition of what is core and periphery could alter the working practices and the relation between the project and implementation projects in hospitals. How do clinical IT departments understand their relation to openEHR?

- The ties requirements have with innovation studies could be deepened, specifically researching how becoming affects the directionality and intentionality of the project's requirements. For example, would less virtual requirements benefit more from an intentional guidance and those which are more virtual benefit from directional guidance (one suppressing their becoming, the other opening it)?

- The previous point has a direct corollary: how do requirements in-becoming affect a project's sustainability? Departing from Ostrom's institutional theory and applying less deterministic concepts would provide a richer understanding of the internalising of actors and the roles they play in sustaining a project. In other words, how does the changing nature of actors (requirements included) influence sustainability?

10.4 Summary

The purpose of this thesis was to question the nature of requirements and the ontological assumptions with which they are often associated. It seemed important to investigate what requirements are, what could they become. Requirements are not only representations of a software's functionality, nor a system which neatly categorises parts and wholes. Requirements also come to represent and mediate an alignment, or disalignment, between one world and another. How this mediation is viewed will ultimately influence what requirements are. If, for example, requirements were normative entities (as they are usually considered in requirements engineering), the process and the 'resulting' world would be differently affected than if requirements were seen as constructed. These views, both the positivist-normative and the constructivist, assume a stability and singularity of the world and of requirements.

To research what requirements are, it is necessary to depart from the underlying assumptions that constrain their nature. To question their nature, this thesis used Deleuze and Guattarian concepts; it is principally their ontology of becoming which has already been applied (in spirit if not in name) to processual studies. The Deleuzian concepts were useful in countering the core tenets of stability, often seen as dominating or as a 'natural' state to which change would eventually submit to. Becoming, in particular, by allowing rhizomes to develop (and thus multiplicities), was fundamental to the search of alternative conceptualisations of requirements.

Open source seemed an appropriate laboratory to look for alternative conceptions of requirements because no agreement has yet been reached over what open source is exactly and how it can influence requirements processes. Open source, thus, provided an amenable setting to the application of unorthodox theories of becoming. openEHR, an open source project in health IT proved an interesting base to theorise on the nature of requirements and their processes. Mainly, its explicit concerns for requirements made for a study that researched their roles and purpose.

The methodological tools chosen reflected the theoretical foundation and the field of enquiry. Beyond a counter-position to stability, no assumptions are made about requirements. The concern to understand requirements and their assemblages is based on the consideration that requirements could be full actors, and not tamed objects. To study these actors, mailing lists and interviews were considered as good sources of data. Mailing lists are recurrent sources in open source studies, while interviews feature prominently in case study based information systems research. Coding played an important role in gathering and relating data together, and helped applying the theoretical concepts to the evidence. Data analysis resulted in three principal themes that were important in sourcing and articulating requirements: identity; territories and contexts; and modes of engagement, supported by two other themes: time/space, and values.

The theme of identity showed the relation between requirements and openEHR's concern for being in the world. Such relation took three prominent forms: building a collective to support in the sourcing and articulation of requirements; the influence requirements can have on the project's identity; and openEHR's position and relevance to the other players in the EHR field.

The theme of territories and contexts questioned the role of space (through time) with requirements. Delineating clear boundaries in the territories to be contextualised is difficult and represents a constant mapping process for openEHR. Requirements are influenced by the moving territories in the world, which in turn, changes openEHR's realities. The constant movement brings to the fore questions of ownership and control of requirements, since their in-becoming nature in difficult to grasp ever.

Because of openEHR's shifting identity and the moving fields which it tries to settle, openEHR has to device ways to make sense of requirements. It does this through two principal, non-exclusive modes: emergence and stabilisation. Both these modes are not judged in a vacuum; one is not necessarily better than the other. openEHR makes use of the two to grow requirements and accommodate them into itself, its identity, and its other requirements. Such a relativistic understanding of requirements is significant: their substance and representations are not obvious any more, nor are they whole or complete, but full of competing potential.

A virtual requirement evolves and is able to perform a number of potential realities. Such requirements do not only describe a machine, or reorder the world deterministically; they enact a multiplicity of realities through their own potential to affect the world. For a requirement, the faculty to be virtual derives from their lack of submission: they and their assemblages cannot be bound easily, or do not fit well with the intentions to which they have been made. In this sense, virtual requirements have a number of ways they can be made actual, and therefore can perform different realities, some potentially more beneficial. The best illustration of a virtual requirement is perhaps to see them as cyborgs (Haraway 1991). Just like cyborgs, their definitions are leaky: they do not belong to any polarity, neither entirely social nor entirely technical; nor are they created through typical means of reproduction, being neither born from a clear process, nor transcendentally pre-existing their creation. They are, like requirements, a middle between machine and man, being both and each one at the same time. Just like the cyborg who (which?) questions humanity, a requirement questions potential and actual realities. It is through its questionable being, both actual and potential, that a requirement can promise the enactment of that potentiality. The enquiry becomes a matter of accommodating those promising requirements that do not (yet) fit inside the actual realities.

Part V

Bibliography and Addenda

References

Aaltonen, A. and N. Tempini (2014). "Everything counts in large amounts: a critical realist case study on data-based production". In: *Journal of Information Technology* 29.1, pp. 97–110.

Ackerman, M. S. (2000). "The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility". In: *Human-Computer Interaction*.

Agarwal, R., C. Anderson, et al. (2013). "If We Offer it, Will They Accept? Factors Affecting Patient Use Intentions of Personal Health Records and Secure Messaging". In: *Journal of Medical Internet Research* 15.2, e43.

Agarwal, R. and C. M. Angst (2009). "Adoption of Electronic Health Records in the Presence of Privacy Concerns: the Elaboration Likelihood Model and Individual Persuasion". In: *MIS Quarterly* 33.2, pp. 339–370.

Akrich, M. (1992). "The De-Scription of Technical Objects". In: *Shaping Technology*. Ed. by W. Bijker and J. Law. The MIT Press.

Allones, J. et al. (2013). "SNOMED CT module-driven clinical archetype management". In: *Journal of Biomedical Informatics*.

AlMarzouq, M. et al. (2005). "Open Source: Concepts, Benefits, and Challenges". In: *Communications of the Association for Information Systems* 16.37, pp. 756–784.

Alvarez, R. (2002). "Confessions of an information worker: a critical analysis of information requirements discourse". In: *Information and Organization* 12.2, pp. 85–107.

Anderson, R. (2008). “Confidentiality and Connecting for Health”. In: *British Journal of General Practice* 58.547, pp. 75–76.

Andrew, B. (2003). “Nurses who left the British NHS for private complementary medical practice: why did they leave? Would they return?” In: *Journal of Advanced Nursing* 41.4, pp. 403–415.

Appan, R. and G. J. Browne (2012). “The Impact of Analyst-Induced Misinformation on the Requirements Elicitation Process”. In: *MIS Quarterly* 36.1, p. 85.106.

Asay, M. (2012). *Has cash corrupted open source?* URL: `http://www.theregister.co.uk/2012/08/24/open_source_entrepreneurship/`.

Asay, M. (2013). “Q&A. Is Open Source Sustainable?” In: *Technology Innovation Management Review* January: Open Source Sustainability.

Atalag, K., H. Y. Yang, and J. Warren (2012). “Assessment of Software Maintainability of openEHR Based Health Information Systems – A Case Study In Endoscopy”. In: *electronic Journal of Health Informatics* 7.1, e3.

Avgerou, C. (2001). “The significance of context in information systems and organizational change”. In: *Information Systems Journal* 11.1, pp. 43–63.

Avgerou, C. (2013). “Social Mechanisms for Causal Explanation in Social Theory Based IS Research”. In: *Journal of the Association for Information Systems* 14.8, pp. 399–419.

Bacon, J. (2012). *The Art of Community: Building the New Age of Participation*. 2nd ed. O’Reilly Media.

Bakker, A. R. (2007). “The need to know the history of the use of digital patient data, in particular the EHR”. In: *International Journal of Medical Informatics* 76.5-6, pp. 438–441.

Barad, K. (2013). “On Touching—The Inhuman That Therefore I Am”. In: *differences* 23.3, pp. 206–223.

Barad, K. (2007). *Meeting the Universe Halfway: Quantum Physics And the Entanglement of Matter And Meaning*. Duke University Press.

Barber, N. et al. (2011). “User-centered healthcare IT: meaningful or meaningless?” In: *MIE*. Oslo, Norway.

Barrett, M. and E. Oborn (2010). “Boundary object use in cross-cultural software development teams”. In: *Human Relations* 63.8, pp. 1199–1221.

Baskerville, R. L. and A. Lee (2003). “Generalizing Generalizability in Information Systems Research”. In: *Information Systems Research* 14.3, pp. 221–243.

Baskerville, R. L. and J. Pries-Heje (2004). “Short cycle time systems development”. In: *Information Systems Journal* 14.3, pp. 237–264.

Batifoulier, P., L. Braddock, and J. Latsis (2012). “Priority Setting in Health Care: From Arbitrariness to Societal Values”. In: *Journal of Institutional Economics* 9.01, pp. 61–80.

Bauwens, M. (2009). “Class and capital in peer production”. In: *Capital & Class* 33.1, pp. 121–141.

Beauvoir de, S. (1976). *Le deuxième sexe II*. Gallimard.

Beck, K. (1999). “Embracing Change with Extreme Programming”. In: *Computer* 32.10, pp. 70–77.

Bell, T. E., D. C. Bixler, and M. E. Dyer (1977). “An Extendable Approach to Computer-Aided Software Requirements Engineering”. In: *IEEE Transactions on Software Engineering* SE-3.1, pp. 49–60.

Bell, T. and T. Thayer (1976). “Software requirements: Are they really a problem?” In: *Proceedings of the 2nd international conference on Software engineering*, pp. 61–68.

Benbasat, I., D. K. Goldstein, and M. Mead (1987). “The Case Research Strategy in Studies of Information Systems”. In: *MIS Quarterly* 11.3, pp. 369–386.

Bencomo, N. et al. (2010). “Requirements Reflection: Requirements as Runtime Entities”. In: *The 32nd ACM/IEEE International Conference*. New York, USA: ACM Press, p. 199.

Benkler, Y. (2011). *The penguin and the leviathan: How cooperation triumphs over self-interest*. Crown Business.

Bergquist, M. and J. Ljungberg (2001). “The power of gifts: organizing social relationships in open source communities”. In: *Information Systems Journal* 11.4, pp. 305–320.

Bergson, H. L. (2010). *Matter and Memory*. Digireads.com.

Berry, D. M. (2011). “Software requirements and design: the work of Michael Jackson by Bashar Nuseibeh and Pamela Zave”. In: *ACM SIGSOFT Software Engineering Notes* 36.2, pp. 39–40.

Beunza, D. and D. Stark (2012). “From dissonance to resonance: cognitive interdependence in quantitative finance”. In: *Information Technology and People* 41.3, pp. 383–417.

Biddle, J. (2013). *Interview with Chris DiBona of Google*. URL: `http://opensource.com/business/13/10/interview-chris-dibona`.

Biernacki, R. (2012). *Reinventing Evidence in Social Inquiry: Decoding Facts and Variables Cultural Sociology*. Palgrave Macmillan.

Bisbal, J., D. Lawless, and J. Grimson (1999). “Legacy information systems: Issues and directions”. In: *IEEE Software* September/October.

Black, A. D. et al. (2011). “The Impact of eHealth on the Quality and Safety of Health Care: A Systematic Overview”. In: *PLoS Medicine* 8.1.

Boehm, B. (2002). “Requirements that handle IKIWISI, COTS, and rapid change”. In: *Computer* 33.7, pp. 99–102.

Boehm, B. W. (1984a). “Verifying and Validating Software Requirements and Design Specifications”. In: *IEEE Software* 1.1, pp. 75–88.

Boehm, B. (1984b). “Software Engineering Economics”. In: *Software Engineering, IEEE Transactions on* SE-10.1, pp. 4–21.

— (1986). “A Spiral Model of Software Development and Enhancement”. In: *ACM SIGSOFT Software Engineering Notes* 11.4, pp. 14–24.

— (1998). “A Stakeholder Win-Win Approach to Software Engineering Education”. In: *Annals of Software Engineering* 6.1-4, pp. 295–321.

Boltanski, L. and L. Thevenot (2006). *On Justification: Economies of Worth*. Princeton Studies in Cultural Sociology. Princeton University Press.

Bonaccorsi, A., S. Giannangeli, and C. Rossi (2006). “Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry”. In: *Management Science* 52.7, pp. 1085–1098.

Bonaccorsi, A. and C. Rossi (2006). “Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business”. In: *Knowledge, Technology & Policy* 18.4, pp. 40–64.

Boness, K., A. Finkelstein, and R. Haririson (2011). “A method for assessing confidence in requirements analysis”. In: *Information and Software Technology* 53.

Bourdieu, P. (1992). *The Logic of Practice*. Polity Press.

Bowker, G. and S. L. Star (2000). *Sorting Things Out*. The MIT Press.

Brooks, F. (1995). *The Mythical Man-Month - Essays on Software Engineering*. Anniversary edition. Addison-Wesley Professional.

Burton, L. C., G. F. Anderson, and I. W. Kues (2004). “Using Electronic Health Records to Help Coordinate Care”. In: *The Milbank Quarterly* 82.3, pp. 457–481.

Buss, J. A. (1981). “The politics of Software Engineering”. In: *The 9th annual ACM SIGUCCS conference*. New York, New York, USA: ACM, pp. 18–20.

Callon, M. (2002). “Writing and (Re)writing Devices as Tools for Managing Complexity”. In: *Complexities: Social Studies of Knowledge Practices*. Ed. by J. Law and A. Mol. Duke University Press.

Callon, M., P. Lascoumes, and Y. Barthe (2011). *Acting in an Uncertain World: An Essay on Technical Democracy*. Inside Technology. The MIT Press.

Callon, M., Y. Millo, and F. Muniesa, eds. (2007). *Market devices*. Blackwell Publishing.

Capra, E., C. Francalanci, and F. Merlo (2008). “An Empirical Study on the Relationship among Software Design Quality, Development Effort, and Governance in Open Source Projects”. In: *IEEE Transactions on Software Engineering* 34.6, pp. 765–782.

Cave, W. C. and A. B. Salisburi (1978). “Controlling the Software Life Cycle—The Project Management Task”. In: *IEEE Transactions on Software Engineering* SE-4.4, pp. 326–334.

Céret, E. et al. (2013). “A taxonomy of design methods process models”. In: *Information and Software Technology* 55.5, pp. 795–821.

Champeaux, D. de et al. (1990). “Structured analysis and object oriented analysis”. In: *ACM SIGPLAN Notices*. ACM, pp. 135–139.

Chan, L. W. C. et al. (2011). “A SNOMED supported ontological vector model for subclinical disorder detection using EHR similarity”. In: *Engineering Applications of Artificial Intelligence* 24.8, pp. 1398–1409.

Chand, D. R. and S. B. Yadav (1980). “Logical construction of software”. In: *Communications of the ACM* 23.10, pp. 546–555.

Chang, C. K. et al. (2001). “Function-class decomposition: a hybrid software engineering method”. In: *Computer* 34.12, pp. 87–93.

Chapin, N. (1978). “Some structured analysis techniques”. In: *ACM SIGMIS Database* 10.3, pp. 16–23.

Charmaz, K. C. (2006). *Constructing Grounded Theory: A Practical Guide through Qualitative Analysi*. Sage.

Checkland, P. (1985a). “Achieving ’Desirable and Feasible’ Change: An Application of Soft Systems Methodology”. In: *Journal of the Operational Research Society*, p. 821.

Checkland, P. (1985b). “From Optimizing to Learning: A Development of Systems Thinking for the 1990s”. In: *The Journal of the Operational Research Society* 36.9, p. 757.

— (2000). “Soft systems methodology: a thirty year retrospective”. In: *Systems Research and Behavioral Science*, pp. 11–58.

Checkland, P. and J. Poulter (2006). *Learning for Action: A Short Definitive Account of Soft Systems Methodology and its use for Practitioners, Teachers, and Students*. Wiley.

Cheliotis, G. (2009). “From Open Source to Open Content: Organization, Licensing and Decision Processes in Open Cultural Production”. In: *Decision Support Systems* 47.3, pp. 229–244.

Chen, R. (2009). *Towards Interoperable and Knowledge-Based Electronic Health Records Using Archetype Methodology*.

Chiasson, M. W. and E. Davidson (2004). “Pushing the contextual envelope: developing and diffusing IS theory for health information systems research”. In: *Information and Organization* 14.3, pp. 155–188.

Chopra, A. K. and M. P. Singh (2012). “Interaction-Oriented Software Engineering: Concepts and Principles”. In:

Churchman, C. W. (1971). *The Design of Inquiring Systems*. Basic Books.

Ciborra, C. (2002a). “Bricolage”. In: *The Labyrinths of Information: Challenging the Wisdom of Systems*. Oxford University Press, pp. 29–53.

— (2002b). *The Labyrinths of Information: Challenging the Wisdom of Systems*. Oxford University Press.

Ciborra, C. U. and G. F. Lanzara (1994). “Formative contexts and information technology: Understanding the dynamics of innovation in organizations”. In: *Accounting, Management and Information Technologies* 4.2, pp. 61–86.

Clegg, C. and C. Shepherd (2007). “‘The biggest computer programme in the world…ever!’: time for a change in mindset?” In: *Journal of Information Technology* 22.3, pp. 212–221.

Coad, P. and E. Yourdon (1990). *Object-Oriented Analysis*. 2nd. Prentice Hall.

Cohn, M. (2004). “User stories applied: For agile software development”. In: *Addison Wesley Signature Series*.

Cohn, M. L., S. E. Sim, and C. P. Lee (2009). “What Counts as Software Process? Negotiating the Boundary of Software Work Through Artifacts and Conversation”. In: *Computer Supported Cooperative Work (CSCW)* 18.5-6, pp. 401–443.

Coiera, E. (2007). “Lessons from the NHS National Programme for IT”. In: *Medical Journal of Australia* 186.1.

Coiera, E. (2009). “Building a National Health IT System from the Middle Out”. In: *Journal of the American Medical Informatics Association* 16.3, pp. 271–273.

Coiera, E., J. Aarts, and C. Kulikowski (2012). “The Dangerous Decade”. In: *Journal of the American Medical Information Association* 19.1, pp. 2–5.

Colazo, J. A. and Y. Fang (2010). "Following the Sun: Temporal Dispersion and Performance in Open Source Software Project Teams." In: *Journal of the Association for Information Systems* 11.11, pp. 684–707.

Coleman, G. (2009). "Code Is Speech: Legal Tinkering, Expertise, and Protest Among Free and Open Source Software Developers". In: *Cultural Anthropology* 24.3.

— (2012). *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton University Press.

Conboy, K. (2009). "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development". In: *Information Systems Research* 20.3, pp. 329–354.

Cornford, T., M. Shaikh, and C. Ciborra (2010). "Hierarchy, Laboratory and Collective: Unveiling Linux as Innovation, Machination and Constitution." In: *Journal of the Association for Information Systems* 11.12, pp. 809–837.

Coulombe, M. (2012). *Petite philosophie du zombie : Ou comment penser l'horreur*. Presses Universitaires de France.

Crotty, M. (1998). *The Foundations of Social Research The Foundations of Social Research - Meaning and Perspective in the Research Process*.

Crowston, K. and J. Howison (2005). "The Social Structure of Free and Open Source Software Development". In: *First Monday* 10.

Curtis, B., H. Krasner, and N. Iscoe (1988). "A field study of the software design process for large systems". In: *Communications of the ACM* 31.11, pp. 1268–1287.

Curto-Millet, D. (2013). "Sustainability and Governance in Developing Open Source Projects as Processes of In-Becoming". In: *Technology Innovation Management Review* January: Open Source Sustainability.

Cusumano, M. and R. Selby (1997). "How Microsoft builds software". In: *Communications of the ACM*.

Daniel, G. W., E. Ewen, et al. (2010). “Efficiency and Economic Benefits of a Payer-Based Electronic Health Record in an Emergency Department”. In: *Academic Emergency Medicine* 17.8, pp. 824–833.

Daniel, S., R. Agarwal, and K. J. Stewart (2013). “The Effects of Diversity in Global, Distributed Collectives: A Study of Open Source Project Success”. In: *Information Systems Research* 24.2, pp. 312–333.

Darking, M. L. and E. A. Whitley (2007). “Towards an Understanding of FLOSS”. In: *Science Studies* 20.2, pp. 13–33.

Davidson, E. (2002). “Technology Frames and Framing: A Socio-Cognitive Investigation of Requirements Determination”. In: *MIS Quarterly* 26.4, pp. 329–358.

Davis, A. M., E. H. Bersoff, and E. R. Comer (1988). “A strategy for comparing alternative software development life cycle models”. In: *IEEE Transactions on Software Engineering* 14.10, pp. 1453–1461.

Davis, A. M. and P. Sitaram (1994). “A concurrent process model of software development”. In: *ACM SIGSOFT Software Engineering Notes* 19.2, pp. 38–51.

De Laat, P. B. (2004). “Evolution of Open Source Networks in Industry”. In: *The Information Society* 20.4, pp. 291–299.

— (2012). “Coercion or empowerment? Moderation of content in Wikipedia as ‘essentially contested’ bureaucratic rules”. In: *Ethics and Information Technology* 14.2.

— (2014). “From open-source software to Wikipedia: ‘Backgrounding’ trust by collective monitoring and reputation tracking”. In: *Ethics and Information Technology* 16.2.

DeLanda, M. (2006). *A New Philosophy of Society: Assemblage Theory and Social Complexity*. Bloomsbury.

Deleuze, G. (2003). *Pourparlers 1972-1990*. Les Éditions de Minuit.

— (2004). *Difference and Repetition*. Bloomsbury Academic.

— (2005a). “Hume”. In: *Pure Immanence: Essays on A Life*. Zone Books.

Deleuze, G. (2005b). “Immanence: A Life”. In: *Pure Immanence: Essays on A Life*. Zone Books.

Deleuze, G. and F. Guattari (1987). *A Thousand Plateaus: Capitalism and Schizophrenia*. Continuum International.

— (2004). *Anti-Oedipus: Capitalism and Schizophrenia*. Continuum Impacts.

Deleuze, G. (2006). *The Fold*. Continuum Impacts.

Deleuze, G. and F. Guattari (1991). *Qu’est-ce que la philosophie*. Les Éditions de Minuit.

Derrida, J. (1967). *De la Grammatologie*. Les Éditions de Minuit.

Di Gangi, P. M. and M. Wasko (2009). “Steal my idea! Organizational adoption of user innovations from a user innovation community: A case study of Dell IdeaStorm”. In: *Decision Support Systems* 48.1, pp. 303–312.

Dias, R. D., T. W. Cook, and S. M. Freire (2011). “Modeling healthcare authorization and claim submissions using the openEHR dual-model approach”. In: *BMC Medical Informatics and Decision Making* 11.1, p. 60.

Dinh-Trong, T. T. and J. M. Bieman (2005). “The FreeBSD project: a replication case study of open source development”. In: *IEEE Transactions on Software Engineering* 31.6, pp. 481–494.

Dionysiou, D. D. and H. Tsoukas (2013). “Understanding the (Re)Creation of Routines from Within: A Symbolic Interactionist Perspective”. In: *Academy of Management Review* 38.2, pp. 181–205.

Doganova, L. and M. Eyquem-Renault (2009). “What do business models do?: Innovation devices in technology entrepreneurship”. In: 38.10, pp. 1559–1570.

Doolin, B. and L. McLeod (2012). “Sociomateriality and boundary objects in information systems development”. In: *European Journal of Information Systems* 21.5, pp. 570–586.

Dreyfus, H. (2007). “Why Heideggerian AI failed and how fixing it would require making it more Heideggerian”. In: *Artificial Intelligence* 13.2, pp. 117–126.

Duboc, L., E. Letier, and D. S. Rosenblum (2013). “Systematic Elaboration of Scalability Requirements through Goal-Obstacle Analysis”. In: *IEEE Transactions on Software Engineering* 39.1, pp. 119–140.

Faraj, S., S. L. Jarvenpaa, and A. Majchrzak (2011). “Knowledge Collaboration in Online Communities”. In: *Organization Science* 22.5, pp. 1224–1239.

Faraj, S. and S. L. Johnson (2011). “Network Exchange Patterns in Online Communities”. In: *Organization Science* 22.6, pp. 1464–1480.

Farjoun, M. (2010). “Beyond Dualism: Stability and Change as a Duality”. In: *Academy of Management Review* 35.2, pp. 202–225.

Faulkner, P. and J. Runde (2013). “On Sociomateriality”. In: *Materiality and Organizing: Social Interaction in a Technological World*. Ed. by P. M. Leonardi, B. A. Nardi, and J. Kallinikos. Oxford University Press.

Feero, W. G. et al. (2008). “New Standards and Enhanced Utility for Family Health History Information in the Electronic Health Record: An Update from the American Health Information Community’s Family Health History Multi-Stakeholder Workgroup”. In: *Journal of the American Medical Informatics Association* 15.6, pp. 723–728.

Feyerabend, P. K. (2010). *Against Method*. 4th. Verso.

Finkelstein, A. (2012). “Requirements and Relationships: A Foreword”. In: *Software and Systems Traceability*. Ed. by J. Cleland-Huang, O. Gotel, and A. Zisman. London: Springer-Verlag.

Fishenden, J. (2010). *Farewell then NPfIT across the health service. But without learning longer-term lessons, will locally-orientated IT development in the NHS just be going back to the future?* URL: `http://eprints.lse.ac.uk/39719/`.

Fitzgerald, B. (2006). “The Transformation of Open Source Software”. In: *MIS Quarterly* 30.3, pp. 587–599.

Ford, E. W., N. Menachemi, and M. T. Phillips (2006). “Predicting the Adoption of Electronic Health Records by Physicians: When Will Health Care be Paperless?” In: *Journal of the American Medical Informatics Association* 13.1, pp. 106–112.

Fraser, S. et al. (2009). “The Role of Judgment in Software Estimation”. In: *2009 31st International Conference on Software Engineering - Companion Volume*. IEEE, pp. 13–17.

Freeman, S. (2012). “User freedom or user control?: The discursive struggle in choosing among Free/Libre Open Source tools in the Finnish public sector”. In: *Information Technology and People* 25.1, pp. 103–128.

Freud, S. (2003). *The Uncanny*. Penguin Modern Classics.

Friedman, D. J. and R. G. Parrish (2010). “The population health record: concepts, definition, design, and implementation”. In: *Journal of the American Medical Informatics Association* 17.4, pp. 359–366.

Galloway, A. R. (2012). *The Interface Effect*. Polity.

Garde, S. et al. (2007). “Expressing clinical data sets with openEHR archetypes: A solid basis for ubiquitous computing”. In: *International Journal of Medical Informatics* 76, S334–S341.

Gause, D. C. and G. M. Weinberg (1989). “Exploring Requirements: Quality before Design”. In:

Gawande, A. (2011). *The Checklist Manifesto: How To Get Things Rights*. Profile Books.

German, M. (2005). “Software Engineering Practices in GNOME”. In: *Perspectives on Free and Open Source Software*. Ed. by J. Feller et al. The MIT Press, pp. 211–225.

Giddens, A. (1984). *The Constitution of Society*. Polity Press.

Gillies, A. C. and I. Patel (2009). “IT and the NHS: Investigating Different Perspectives of IT using Soft Systems Methodology”. In: *Studies in Ethics, Law, and Technology* 3.2.

Glaser, B. G. and A. L. Strauss (1967). *Discovery of Grounded Theory: Strategies for Qualitative Research*. AldineTransaction.

Greenhalgh, T. et al. (2010). “Adoption and non-adoption of a shared electronic summary record in England: a mixed-method case study”. In: *BMJ* 340.jun16 4, pp. c3111–c3111.

Greenspan, S., J. Mylopoulos, and A. Borgida (1994). “On formal requirements modeling languages: RML revisited”. In: *16th International Conference on Software Engineering*. IEEE, pp. 135–147.

Grogan, J. (2006). “EHRs and Information Availability: Are You At Risk?” In: *Health Management Technology* 27.5.

Group, S. (1994). *CHAOS Chronicles*. Tech. rep.

— (2003). *CHAOS Chronicles v3.0*. Tech. rep.

Guattari, F. (1995). *Chaosmosis: An Ethico-aesthetic Paradigm*. Power Institute of Fine Arts.

Hacking, I. (1999). *The Social Construction of What?*

Halamka, J. D., K. D. Mandl, and P. C. Tang (2007). “Early Experiences with Personal Health Records”. In: *Journal of the American Medical Informatics Association* 15.1, pp. 1–7.

Ham, C., R. Kipping, and H. McLeod (2003). “Redesigning Work Processes in Health Care: Lessons from the National Health Service”. In: *The Milbank Quarterly* 81.3, pp. 415–439.

Haraway, D. (1991). *Simians, Cyborgs and Women: The Reinvention of Nature*. 2nd. Free Association Books.

Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics*. re.press.

Hartman-Kroah, G., J. Corbet, and A. McPherson (2009). *Linux Kernel Development*. The Linux Foundation.

Hassol, A. (2004). “Patient Experiences and Attitudes about Access to a Patient Electronic Health Care Record and Linked Web Messaging”. In: *Journal of the American Medical Informatics Association* 11.6, pp. 505–513.

Hayrinen, K., K. Saranto, and P. Nykanen (2008). “Definition, Structure, Content, Use and Impacts of Electronic Health Records: a Review of the Research Literature”. In: *International Journal of Medical Informatics* 77.5, pp. 291–304.

Häyrinen, K., K. Saranto, and J. Lammintakanen (2010). “Evaluation of electronic nursing documentation—Nursing process model and standardized terminologies as keys to visible and transparent nursing". In: *International Journal of Medical Informatics* 79.8, pp. 554–564.

Hendy, J. et al. (2007). "Implementing the NHS information technology programme: qualitative study of progress in acute trusts". In: *BMJ* 334.7608, pp. 1360–1360.

Heninger, K. L. (1980). "Specifying Software Requirements for Complex Systems: New Techniques and Their Application". In: *IEEE Transactions on Software Engineering* SE-6.1, pp. 2–13.

Hernes, T. (2014). *A Process Theory of Organisation*. Oxford University Press.

Hertzum, M. (2004). "Small-Scale Classification Schemes: A Field Study of Requirements Engineering". In: *Computer Supported Cooperative Work (CSCW)* 13.1, pp. 35–61.

Hillenius, G. (2012). *Contribution of open source to Europe's economy: 450 billion per year*. URL: `https://joinup.ec.europa.eu/news/contribution-open-source-europes-economy-450-billion-year`.

Hinds, P. and D. Bailey (2003). "Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams". In: *Organization Science* 14.6, pp. 615–632.

Hippel von, E. (2007). "Horizontal innovation networks - by and for users". In: *Industrial and corporate change* 16.2, pp. 293–315.

Hippel von, E., N. Franke, and R. Prügl (2009). ""Pyramiding: Efficient search for rare subjects"". In: 38.9, pp. 1397–1406.

Hippel von, E. and G. Krogh von (2006). "Free revealing and the private-collective model for innovation incentives". In: *R and D Management* 36.3, pp. 295–306.

Hirschheim, R. and H. K. Klein (1989). "Four paradigms of information systems development". In: *Communications of the ACM* 32.10, pp. 1199–1216.

Holmström, J. and S. Sawyer (2010). "Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements". In: *European Journal of Information Systems* 20.1, pp. 34–47.

Howison, J. and K. Crowston (2014). "Collaboration Through Open Superposition: a Theory of the Open Source Way". In: *MIS Quarterly* 38.1.

Humphreys, B. L. (2010). “Electronic Health Record Meets Digital Library: A New Environment for Achieving an Old Goal”. In: *Journal of the American Medical Informatics Association : JAMIA* 7.5, p. 444.

Iivari, J. and N. Iivari (2011). “Varieties of user-centredness: an analysis of four systems development methods”. In: *Information Systems Journal* 21.2, pp. 125–153.

Iivari, N. (2009a). “Empowering the users? A critical textual analysis of the role of users in open source software development”. In: *AI & Society*.

— (2010). “Discursive construction of ’user innovations’ in the open source software development context”. In: *Information and Organization* 20.2, pp. 111–132.

Iivari, N. (2009b). ““Constructing the users” in open source software development: An interpretive case study of user participation”. In: *Information Technology and People* 22.2, pp. 132–156.

Ingram, D. and S. S. Arikan (2013). “The Evolving Role of Open Source Software in Medicine and Health Services”. In: *Technology Innovation Management Review* January: Open Source Sustainability.

Ison, R. et al. (2014). “Insights into Operationalizing Communities of Practice from SSM-Based Inquiry Processes”. In: *Systemic Practice and Action Research* 27.2, pp. 91–113.

Jackson, M. (1978). “Information Systems: Modelling, Sequencing and Transformations”. In: *ICSE ’78: Proceedings of the 3rd international conference on Software engineering*. IEEE Press.

Jackson, M. (1983). *System Development*. Prentice Hall International.

— (1995a). *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices*. Addison Wesley.

— (1995b). “The world and the machine”. In: *ICSE ’95: Proceedings of the 17th international conference on Software engineering*, pp. 283–292.

— (2008). “Problem frames and software engineering”. In: *Expert Systems* 25.1, pp. 7–8.

Jackson, M. (2012). “Aspects of abstraction in software development”. In: *Software & Systems Modeling* 11.4, pp. 495–511.

Jackson, M. and P. Zave (1995). *Deriving specifications from requirements: An example*.

Jarke, M. et al. (2011). “The brave new world of design requirements”. In: *Information Systems* 36.7, pp. 992–1008.

Jason, G. (1989). “The role of error in computer science”. In: *Philosophia* 19.4, pp. 403–416.

Johnson, S. B. et al. (2007). “An Electronic Health Record Based on Structured Narrative”. In: *Journal of the American Medical Informatics Association* 15.1, pp. 54–64.

Jorgensen, N. (2001). “Putting it all in the trunk: incremental software development in the FreeBSD open source project”. In: *Information Systems Journal* 11.4, pp. 321–336.

Juristo, N., A. M. Moreno, and A. Silva (2002). “Is the European industry moving toward solving requirements engineering problems?” In: *IEEE Software* 19.6, pp. 70–77.

Kallinikos, J. (2005). “The order of technology: Complexity and control in a connected world”. In: *Information and Organization* 15.3, pp. 185–202.

— (2006). *The Consequences of Information: Institutional Implications of Technological Change*. Edward Elgar.

— (2009). “On the computational rendition of reality: artefacts and human agency”. In: *Organization* 16.2, p. 183.

— (2011). *Governing Through Technology: Information Artefacts and Scoial Practice*. Palgrave Macmillan.

Kalra, D. (1994). “Medicine in Europe: Electronic health records: the European scene”. In: *BMJ* 309.6965, pp. 1358–1361.

Kalra, D. et al. (2006). “Confidentiality and consent in medical research: Confidentiality of personal health information used for research”. In: *BMJ : British Medical Journal* 33.

Ke, W. and P. Zhang (2010). “The Effects of Extrinsic Motivations and Satisfaction in Open Source Software Development.” In: *Journal of the Association for Information Systems* 11.12, pp. 784–808.

Kelty, C. M. (2004). “Culture’s Open Sources: Software, Copyright, and Cultural Critique”. In: *Anthropological Quaterly* 77.3.

Kho, A. N. et al. (2012). “Use of diverse electronic medical record systems to identify genetic risk for type 2 diabetes within a genome-wide association study”. In: *Journal of the American Medical Informatics Association* 19.2, pp. 212–218.

Klecun, E. (2008). “Bringing lost sheep into the fold: questioning the discourse of the digital divide”. In: *Information Technology and People* 21.3, pp. 267–282.

Klein, H. K. and M. D. Myers (1999). “A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems”. In: *MIS Quarterly* 23.1, pp. 67–94.

Kluge, E.-H. W. (2004). “Informed consent and the security of the electronic health record (EHR): some policy considerations”. In: *International Journal of Medical Informatics* 73.3, pp. 229–234.

Krogh von, G. et al. (2012). “Carrots and rainbows: motivation and social practice in open source software development”. In: *MIS Quarterly* 36.2.

Kruchten, P. (2010). “Software architecture and agile software development: a clash of two cultures?” In: *The 32nd ACM/IEEE International Conference*. New York, New York, USA: ACM, pp. 497–498.

Kuijpers, W. et al. (2013). “A Systematic Review of Web-Based Interventions for Patient Empowerment and Physical Activity in Chronic Diseases: Relevance for Cancer Survivors”. In: *Journal of Medical Internet Research* 15.2, e37.

Lakhani, K. and R. Wolf (2005). “Why hackers do what they do: Understanding motivation and effort in free/open source software projects”. In: *Perspectives on Free and Open Source Software*. Ed. by J. Feller et al. The MIT Press, pp. 1–27.

Lamsweerde, A. van (2000). “Requirements in the Year 00: A Roadmap”. In: *ICSE ’00: Proceedings of the 22nd international conference on Software engineering* 00, pp. 5–19.

Lamsweerde, A. van (2008). “Requirements engineering: from craft to discipline”. In: *SIGSOFT ’08/FSE-16 Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pp. 238–249.

— (2009a). “From Worlds to Machines”. In: *A Tribute to Michael Jackson*.

— (2009b). *Requirements Engineering - From System Goals to UML Models to Software Specifications*. John Wiley & Sons.

Langley, A. et al. (2013). “Process Studies of Change in Organization and Management: Unveiling Temporality, Activity, and Flow”. In: *Academy of Management Journal* 56.1, pp. 1–13.

Lanham, H. J., L. K. Leykum, and R. R. McDaniel (2012). “Same organization, same electronic health records (EHRs) system, different use: exploring the linkage between practice member communication patterns and EHR use patterns in an ambulatory care setting”. In: *Journal of the American Medical Informatics Association* 19.3, pp. 382–391.

Laplante, P. A. and C. J. Neill (2004). ““The Demise of the Waterfall Model Is Imminent” and Other Urban Myths”. In: *Queue* 1.10, p. 10.

Latour, B. (2013). “Biography of an inquiry: On a book about modes of existence”. In: *Social Studies of Science* 43.2, pp. 287–301.

Latour, B., G. Harman, and P. Erdélyi (2011). *The Prince and the Wolf: Latour and Harman at the LSE*. Zero Books.

Latour, B. (2004). *Politics of Nature: How to Bring the Sciences into Democracy*. Harvard University Press.

— (2005). *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford University Press.

Laurent, P. and J. Cleland-Huang (2010). “Requirements-Gathering Collaborative Networks in Distributed Software Projects”. In: *Requirements: Communication, Understanding and Softskills, 2009 Collaboration and Intercultural Issues on*, pp. 26–30.

Laurent, P. and J. Cleland-Huang (2009). “Lessons learned from open source projects for facilitating online requirements processes”. In: *REFSQ ’09 Proceed-*

ings of the 15th International Working Conference on Requirements Engineering: Foundation for Software Quality 5512, pp. 240–255.

Law, J. (2002). *Aircraft Stories*. Duke University Press.

— (2004). *After-Method: Mess in Social Science Research*. Routledge.

Lehman, G. (2011). “Interpretivism, postmodernism and nature: Ecological conversations”. In: *Philosophy & Social Criticism* 37.7, pp. 795–821.

Leonardi, P. M. (2013). “Theoretical foundations for the study of sociomateriality”. In: *Information and Organization* 23.2, pp. 59–76.

Leonardi, P. M., B. A. Nardi, and J. Kallinikos, eds. (2013). *Materiality and Organizing: Social Interaction in a Technological World*. Oxford University Press.

Lerner, J. and J. Tirole (2002). “Some Simple Economics of Open Source”. In: *Journal of Industrial Economics* 50.2, pp. 197–234.

— (2005). “The Economics of Technology Sharing: Open Source and Beyond”. In: *The Journal of Economic Perspectives* 19.2, pp. 99–120.

Leslie, H. and S. Heard (2006). “Archetypes 101”. In: *HIC Bridging the Digital Divide: Clinician, consumer and computer*. Ed. by J. Westbrook and J. Callen.

Lessig, L. (2006). *Code: version 2.0*. Basic Books.

Letier, E. and A. van Lamsweerde (2004). “Reasoning about partial goal satisfaction for requirements and design engineering”. In: *FSE ’04: Proceedings of the 12th ACM International Symp. on the Foundations of Software Engineering* 29.6, pp. 53–62.

Levy, S. (2010). *Hackers: Heroes of the Computer Revolution*. 25th anniversary edition. O’Reilly Media.

Lewis, R. Q. (2005). “Implementing a national strategy for patient safety: lessons from the National Health Service in England”. In: *Quality and Safety in Health Care* 14.2, pp. 135–139.

Lin, A. and T. Cornford (2000). *Sociotechnical Perspectives on Emergence Phenomena*.

López-Nores, M. et al. (2012). “The iCabiNET system: Harnessing Electronic Health Record standards from domestic and mobile devices to support better medication adherence”. In: *Decision Support Systems* 34.1, pp. 109–116.

Lubars, M., C. Potts, and C. Richter (1993). “A review of the state of the practice in requirements modeling”. In: *IEEE International Symposium on Requirements Engineering*. IEEE, pp. 2–14.

Luhmann, N. (2013). *Introduction to Systems Theory*. Ed. by D. Baecker. Polity Press.

Lyytinen, K. (1987). “Different perspectives on information systems: problems and solutions”. In: *ACM Computing Surveys (CSUR)* 19.1, pp. 5–46.

Mackenzie, A. (2006). *Cutting Code: Software and Sociality Digital Formations*. Peter Lang Publishing.

Malabou, C. (2012). *The Ontology of the Accident: An Essay on Destructive Plasticity*. Polity Press.

Maldonado, J. A. et al. (2012). “Using the ResearchEHR platform to facilitate the practical application of the EHR standards”. In: *Journal of Biomedical Informatics* 45.4, pp. 746–762.

Mandl, K. D., P. Szolovits, and S. I. Kohane (2001). “Public standards and patients’ control: how to keep electronic medical records accessible but private Commentary: Open approaches to electronic patient records Commentary: A patient’s viewpoint”. In: *BMJ* 322.7281, pp. 283–287.

Markus, M. L. (1983). “Power, politics, and MIS implementation”. In: *Communications of the ACM* 26.6, pp. 430–444.

Massumi, B. (1992). *A User’s Guide to Capitalism and Schizophrenia: Deviations from Deleuze and Guattari*. Swerve Edition. The MIT Press.

Mathiassen, L., T. Saarinen, et al. (2007). “A Contigency Model for Requirements Development”. In: *Journal of the Association for Information Systems* 8.11, pp. 569–597.

Mathiassen, L. and C. Sorensen (2008). “Towards a theory of organizational information services”. In: *Journal of Information Technology* 23.4, pp. 313–329.

Mellis, W., C. Loebbecke, and R. L. Baskerville (2013). “Requirements Uncertainty in Contract Software Development Projects”. In: *The Journal of Computer Information Systems* 53.3.

Mingers, J. and L. White (2010). “A review of the recent contribution of systems thinking to operational research and management science”. In: *European Journal of Operational Research* 207.3, pp. 1147–1161.

Mockus, A., R. Fielding, and J. Herbsleb (2002). “Two Case Studies of Open Source Software Development: Apache and Mozilla”. In: *ACM Transactions on Software Engineering and Methodology* 11, pp. 309–346.

Mol, A. (2012). “Mind your plate! The ontonorms of Dutch dieting”. In: *Social Studies of Science*.

Mol, A. (2002). *The Body Multiple: Ontology in Medical Practice*. Science & Cultural Theory. Duke University Press.

— (2008). *The Logic of Care: Health and the Problem of Patient Choice*. 1st ed. Routledge.

Monteiro, E. et al. (2012). “From Artefacts to Infrastructures”. In: *Computer Supported Cooperative Work (CSCW)* 22.4-6, pp. 575–607.

Munson, J. B. (1981). “Special Feature: Software Maintainability: A Practical Concern for Life-Cycle Costs”. In: *Computer* 14.11, pp. 103–109.

Myers, M. D. and H. K. Klein (2011). “A Set of Principles for Conducting Critical Research in Information Systems”. In: *MIS Quarterly* 35.1.

Mylopoulos, J., L. Chung, and E. Yu (1999). “From object-oriented to goal-oriented requirements analysis”. In: *Communications of the ACM* 42.1, pp. 31–37.

Nilsson, A. and T. L. Wilson (2012). “Reflections on Barry W. Boehm’s “A spiral model of software development and enhancement””. In: *International Journal of Managing Projects in Business* 5.4, pp. 737–756.

Noll, J. and W.-M. Liu (2010). “Requirements elicitation in open source software development: a case study”. In: *FLOSS ’10 Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*.

Nuseibeh, B., J. Kramer, and A. Finkelstein (2003). “ViewPoints: meaningful relationships are difficult!” In: *25th International Conference on Software Engineering, 2003. Proceedings.* IEEE, pp. 676–681.

Nuseibeh, B. (2010). “Abuse Frames: Inferring Security Requirementsfrom Anti-Requirements”. In: *Software Requirements and Design: The Work of Michael Jackson*. Ed. by B. Nuseibeh and P. Zave. Chatham, New Jersey: Good Friends Publishing, p. 343.

Nuseibeh, B. and S. Easterbrook (2000). “Requirements Engineering: A Roadmap”. In: *ICSE ’00: Proceedings of the Conference on The Future of Software Engineering*, pp. 35–46.

Okoli, C. and W. Oh (2007). “Investigating recognition-based performance in an open content community: A social capital perspective”. In: *Information and Management* 44.3, pp. 240–252.

O’Mahony, S. and L. Dahlander (2011). “Progressing to the Center: Coordinating Project Work”. In: *Organization Science* 22.4, pp. 961–979.

Orlikowski, W. J. (1996). “Improvising Organizational Transformation Over Time: A Situated Change Perspective”. In: *Information Systems Research* 7.1, pp. 63–92.

— (2000). “Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations”. In: *Organization Science* 11.4, pp. 404–428.

Orlikowski, W. J. and N. Levina (2009). “Understanding Shifting Power Relations within and across Organizations: A Critical Genre Analysis”. In: *Academy of Management Journal* 52.4, pp. 672–703.

Orlikowski, W. J. and S. V. Scott (2013). “What Happens When Evaluation Goes Online? Exploring Apparatuses of Valuation in the Travel Sector”. In: *Organization Science*.

Ostrom, E. (1987). “Institutional arrangements for resolving the commons dilemma: some contending approaches”. In: *Rethinking Institutional Analysis and Development: Issues, Alternatives, and Choices - ICS*, pp. 103–139.

— (1990). *Governing the Commons: The Evolution of Institutions for Collective Action Political Economy of Institutions and Decisions*. Cambridge University Press.

— (2010). “Beyond markets and states: polycentric governance of complex economic systems”. In: *American Economic Review* 100.3.

Parnas, D. and P. Clements (1986). “A Rational Design Process: How and Why to Fake It”. In: *IEEE Transactions on Software Engineering* 12.2, pp. 251–256.

Peters, L. J. (1980). “Software representation and composition techniques”. In: *Proceedings of the IEEE*. IEEE, pp. 1085–1093.

Petrakaki, D. et al. (2011). “The Role of Technology in Shaping the Professional Future of Community Pharmacists: the Case of the Electronic Prescription Service in the English NHS”. In: *Researching the future in information systems*. Ed. by M. Chiasson et al. IFIP Advances in Information and Communication Technology.

Pickering, A. (2011). “Cyborg Spirituality”. In: *Medical History* 55.3, p. 349.

Pickering, A. and K. Guzik (2009). *The Mangle in Practice: Science, Society, and Becoming Science & Cultural Theory*. Duke University Press.

Pollock, N. (2005). “When Is a Work-Around? Conflict and Negotiation in Computer Systems Development”. In: *Science, Technology & Human Values* 30.4, pp. 496–514.

— (2012). “Ranking Devices: The Socio-Materiality of Ratings”. In: *Materiality and Organizing: Social Interaction in a Technological World*. Ed. by P. M. Leonardi, B. A. Nardi, and J. Kallinikos. Oxford University Press.

Pollock, N. and S. Hyysalo (2014). “The business of being a user: the role of the reference actor in shaping packaged enterprise system acquisition and development”. In: *MIS Quarterly* 38.2.

Popper, K. (2002). *The Poverty of Historicism*. 2nd. Routledge.

Potts, C. and L. Catledge (1996). “Collaborative conceptual design: A large software project case study”. In: *Computer Supported Cooperative Work (CSCW)* 5.4, pp. 415–445.

Prados-Suárez, B. et al. (2012). “Improving electronic health records retrieval using contexts”. In: *Expert Systems with Applications* 39.10, pp. 8522–8536.

Preda, A. (2007). “Where Do Analysts Come From? the Case of Financial Chartism”. In: *Market devices*. Ed. by M. Callon, Y. Millo, and F. Muniesa. Blackwell Publishing, p. 40.

Pressman, R. S. (1982). *Software Engineering: A Practitioner’s Approach*. McGraw-Hill.

Quattrone, P. (2009). “Books to be Practiced”. In: *Accounting, Organizations and Society* 34, pp. 85–118.

Raccoon, L. B. S. (1997). “Fifty years of progress in software engineering”. In: *ACM SIGSOFT Software Engineering Notes* 22.1, pp. 88–104.

Rajeev, D. et al. (2010). “Development of an electronic public health case report using HL7 v2.5 to meet public health needs”. In: *Journal of the American Medical Informatics Association* 17.1, pp. 34–41.

Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O’Reilly Media.

Ricœur, P. (2006). *On Translation*. Ed. by R. Kearney. Routledge.

Robbins, J. and J. Feller (2005). “Adopting OSSE Practices by Adopting OSSE Tools”. In: *Perspectives on Free and Open Source Software*. Ed. by B. Fitzgerald, S. A. Hissam, and K. R. Lakhani. The MIT Press, pp. 245–264.

Robbins, P. and S. A. Moore (2013). “Ecological anxiety disorder: diagnosing the politics of the Anthropocene”. In: *Cultural Geographies* 20.3.

Robertson, A., K. Cresswell, et al. (2010). “Implementation and adoption of nationwide electronic health records in secondary care in England: qualitative analysis of interim results from a prospective national evaluation”. In: *BMJ* 341.sep01 3, pp. c4564–c4564.

Robertson, A., T. Cornford, et al. (2012). “The NHS IT project: more than just a bad dream”. In: *The Lancet* 379.9810, pp. 29–30.

Robertson, S. and J. Robertson (2003). *Mastering the Requirements Process*.

Robinson, B. and F. Wilson (2003). “Soft systems methodology and dialectics in an information environment: a case study of the Battle of Britain”. In: *Systems Research and Behavioral Science* 20.3, pp. 255–268.

Ross, D. T. (1977). “Structured Analysis (SA): A Language for Communicating Ideas”. In: *IEEE Transactions on Software Engineering* SE-3.1, pp. 16–34.

Ross, D. T. and K. E. Schoman (1976). “Structured Analysis for Requirements Definition”. In: *IEEE Transactions on Software Engineering* SE-3.1, pp. 6–15.

Rowen, R. B. (1990). “Software project management under incomplete and ambiguous specifications”. In: *Engineering Management, IEEE Transactions on* 37.1, pp. 10–21.

Royce, W. (1970). “Managing the development of large software systems: concepts and techniques”. In: *ICSE ’87: Proceedings of the 9th international conference on Software Engineering*, pp. 328–338.

Rusovan, S., M. Lawford, and D. Parnas (2005). “Assessing Open Source Software Development”. In: *Perspectives on Free and Open Source Software*. Ed. by J. Feller et al. The MIT Press, pp. 107–121.

Saleem, J. J. et al. (2009). “Exploring the persistence of paper with the electronic health record”. In: *International Journal of Medical Informatics* 78.9, pp. 618–628.

Salomon, R. M. et al. (2010). “Openness of patients’ reporting with use of electronic records: psychiatric clinicians’ views”. In: *Journal of the American Medical Informatics Association* 17.1, pp. 54–60.

Sambamurthy, V. et al. (2012). “How Peripheral Developers Contribute to Open-Source Software Development”. In: *Information Systems Research* 23.1, pp. 144–163.

Savage, M. (2014). “Piketty’s challenge for sociology”. In: *The British Journal of Sociology* 65.4, pp. 591–606.

Scacchi, W. (2002). “Understanding the requirements for developing open source software systems”. In: *IEE Proceedings - Software* 149, pp. 24–39.

— (2007). “Free/open source software development: Recent research results and methods”. In: *Advances in Computers* 69, pp. 243–295.

Scacchi, W. (2009). “Understanding Requirements for Open Source Software”. In: *Design Requirements Workshop-Design Requirements Engineering: A Ten-Year Perspective*, pp. 467–494.

Scacchi, W. et al. (2006). “Understanding Free/Open Source Software Development Processes”. In: *Software Process Improvement and Practice* 11.2, pp. 95–105.

Scarbrough, H., N. Panourgias, and J. Nandhakumar (2012). “The Role of Objects in the Coordination of Knowledge-Intensive Projects: A Study of Computer Games Development”. In: *45th Hawaii International Conference on System Sciences (HICSS)*. IEEE, pp. 4952–4960.

Schmidt, K. and L. Bannon (1992). “Taking CSCW seriously”. In: *Computer Supported Cooperative Work (CSCW)* 1.1-2, pp. 7–40.

— (2013). “Constructing CSCW: The First Quarter Century”. In: *Computer Supported Cooperative Work (CSCW)* 22.4-6, pp. 345–372.

Schultz, M. and T. Hernes (2013). “A Temporal Perspective on Organizational Identity”. In: *Organization Science* 24.1.

Schweik, C. M. and R. C. English (2012). “Internet Success: A Study of Open-Source Software Commons”. In: *Internet Success: A Study of Open-Source Software Commons*.

Shaikh, M. and T. Cornford (2012). “Navigating Open Source Adoption in the Public Sector”. In: *AMCIS 2012 Proceedings*.

Simborg, D. W. (2011). “There is no neutral position on fraud!” In: *Journal of the American Medical Informatics Association* 18.5, pp. 675–677.

Sittig, D. F. and H. Singh (2012). “Electronic Health Records and National Patient-Safety Goals”. In: *New England Journal of Medicine* 367.19, pp. 1854–1860.

Sjögren, E. and C.-F. Helgesson (2007). “The Q(u)ALYfying Hand: Health Economics and Medicine in the Shaping of Swedish Markets for subsidized Pharmaceuticals”. In: *Market Devices*. Ed. by M. Callon, Y. Millo, and F. Muniesa. Blackwell Publishing, pp. 215–240.

Skene, J., D. Lamanna, and W. Emmerich (2004). “Precise Service Level Agreements”. In: *ICSE ’04: Proceedings of the 26th International Conference on Software Engineering*, pp. 179–188.

Smith, D. W. (2012). *Essays on Deleuze*. Edinburgh University Press.

Sneed, H. M. (1989). “The myth of ‘top-down’ software development and its consequences for software maintenance”. In: *Proceedings. Conference on Software Maintenance - 1989*. IEEE, pp. 22–29.

Sommerville, I. (2005). “Integrated requirements engineering: a tutorial”. In: *IEEE Software* 22.1, pp. 16–23.

Stacey, M. and C. Eckert (2003). “Against Ambiguity”. In: *Computer Supported Cooperative Work (CSCW)* 12.2, pp. 153–183.

Stallman, R. (1999). “The GNU Operating System and the Free Software Movement”. In: *Open Sources: Voices from the Open Source Revolution*. O’Reilly.

Star, S. L. (1990). “Power, technology and the phenomenology of conventions: on being allergic to onions”. In: *The Sociological Review* 38.S1, pp. 26–56.

Star, S. L. and A. Strauss (1999). “Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work”. In: *Computer Supported Cooperative Work (CSCW)* 8.1-2, pp. 9–30.

Stewart, B. A. et al. (2010). “A preliminary look at duplicate testing associated with lack of electronic health record interoperability for transferred patients”. In: *Journal of the American Medical Informatics Association : JAMIA* 17.3, p. 341.

Subramanyam, R. and M. Xia (2008). “Free/Libre Open Source Software development in developing and developed countries: A conceptual framework with an exploratory study”. In: *Decision Support Systems* 46.1, pp. 173–186.

Takian, A. and T. Cornford (2012). “NHS information: Revolution or evolution?” In: *International Journal of Research in Marketing* 1.4, pp. 193–198.

Takian, A., D. Petrakaki, et al. (2012). “Building a House on Shifting Sand: Methodological Considerations When Evaluating The Implementation and Adoption of National Electronic Health Record Systems”. In: *BMC Health Services Research* 12.105.

Tallon, P. P. (2007). “A Process-Oriented Perspective on the Alignment of Information Technology and Business Strategy”. In: *Journal of Management Information Systems* 24.3.

Taylor, R. P. (2002). “Order in Pollock’s Chaos”. In: *Scientific American* 287.6.

Teege, G. (2000). “Users as Composers: Parts and Features as a Basis for Tailorability in CSCW Systems”. In: *Computer Supported Cooperative Work (CSCW)* 9.1, pp. 101–122.

Thevenot, L. (2002). “Which Road to Follow? The Moral Complexity of an “Equipped” Humanity”. In: *Complexities: Social Studies of Knowledge Practices*. Ed. by J. Law and A. Mol. Duke University Press, pp. 53–87.

Thompson, M. (2011). “Ontological Shift or Ontological Drift? Reality Claims, Epistemological Frameworks, and Theory Generation in Organization Studies.” In: *Academy of Management Review* 36.4, pp. 754–773.

Todd, R. (2014). *Open source GP project takes off*. URL: `http://ehi.co.uk/news/ehi/8125/open-source-gp-project-takes-off`.

Torre-Diez, I., A. González, and M. López-Coronado (2013). “EHR Systems in the Spanish Public Health National System: The Lack of Interoperability between Primary and Specialty Care”. In: *Journal of Medical Systems* 37.1, p. 9914.

Truex, D., R. L. Baskerville, and J. Travis (2000). “Amethodical systems development: the deferred meaning of systems development methods”. In: *Accounting, Management and Information Technologies* 10.1, pp. 53–79.

Tsoukas, H. and R. Chia (2002). “On Organizational Becoming: Rethinking Organizational Change”. In: *Organization Science* 13.5, pp. 567–582.

Tyworth, M. (2014). “Organizational identity and information systems: how organizational ICT reflect who an organization is”. In: *European Journal of Information Systems* 23.1, pp. 69–83.

Ulrich, W. (2012). “Operational Research and Critical Systems Thinking—an Integrated Perspective. Part 1: or as Applied Systems Thinking”. In: *Journal of the Operational Research Society* 63.9, pp. 1228–1247.

Unertl, K. M., K. B. Johnson, and N. M. Lorenzi (2012). "Health information exchange technology on the front lines of healthcare: workflow factors and patterns of use". In: *Journal of the American Medical Informatics Association* 19.3, pp. 392–400.

Urban, G. L. and E. Hippel von (1988). "Lead User Analyses for the Development of New Industrial Products". In: *Management Science* 34.5, pp. 569–582.

Urquhart, C. (1999). "Themes in early requirements gathering: The case of the analyst, the client and the student assistance scheme". In: *Information Technology and People* 12.1, pp. 44–70.

— (2001). "Analysts and clients in organisational contexts: a conversational perspective". In: *The Journal of Strategic Information Systems* 10.3, pp. 243–262.

— (2012). *Grounded Theory for Qualitative Research: A Practical Guide*. Sage Publications.

Urquhart, C. and W. Fernandez (2013). "Using grounded theory method in information systems: the researcher as blank slate and other myths". In: *Journal of Information Technology* 28.3, pp. 224–236.

Vaast, E., E. J. Davidson, and T. Mattson (2013). "Talking About Technology: the Emergence of a New Actor Category Through New Media". In: *MIS Quarterly* 37.4.

VanderLeest, S. H. and A. Buter (2009). "Escape the waterfall: Agile for aerospace". In: *2009 IEEE/AIAA 28th Digital Avionics Systems Conference (DASC)*. IEEE.

Vest, J. R. and J. Jasperson (2010). "What should we measure? Conceptualizing usage in health information exchange". In: *Journal of the American Medical Informatics Association : JAMIA* 17.3, p. 302.

Water, H. van de, M. Schinkel, and R. Rozier. *Journal of the Operational Research Society - Abstract of article: Fields of application of SSM: a categorization of publications*. URL: `http://www.palgrave-journals.com/doifinder/10.1057/palgrave.jors.2602156`.

Weber, R. (2004). “The Rhetoric of Positivism Versus Interpretivism: A Personal View”. In: *MIS Quarterly* 28.1.

— (2012). “Evaluating and Developing Theories in the Information Systems Discipline”. In: *Journal of the Association for Information Systems* 13.1.

Weick, K. (1996). “Drop your tools: An allegory for organizational studies”. In: *Administrative Science Quarterly* 41.2, pp. 301–313.

Weick, K. E. (1998). “Improvisation as a Mindset for Organizational Analysis”. In: *Organization Science* 9.5.

— (2006). “Faith, Evidence, and Action: Better Guesses in an Unknowable World”. In: *Organization Studies* 27.11.

Weinstein, M. (2006). “TAMS Analyzer: Anthropology as Cultural Critique in a Digital Age”. In: *Social Science Computer Review* 24.1, pp. 68–77.

Weinstock, C. and S. A. Hissam (2005). “Making Ligthning Strike Twice”. In: *Perspectives on Free and Open Source Software*. Ed. by J. Feller et al. The MIT Press, pp. 143–159.

Wenger, E. C. and W. Snyder (2000). “Communities of Practice: The Organizational Frontier”. In: *Harvard Business Review* 78.1.

Winkelman, W. J. (2005). “Patient-Perceived Usefulness of Online Electronic Medical Records: Employing Grounded Theory in the Development of Information and Communication Technologies for Use by Patients Living with Chronic Illness”. In: *Journal of the American Medical Informatics Association* 12.3, pp. 306–314.

Wolff, J. G. (1989). “The management of risk in system development: Project SP and the New Spiral Model”. In: *Software Engineering Journal* 4.3, pp. 134–142.

Wollersheim, D., A. Sari, and W. Rahayu (2009). “Archetype-based electronic health records: a literature review and evaluation of their applicability to health data interoperability and access”. In: *Health Information Management Journal* 38.2, pp. 7–17.

Wu, C.-G., J. H. Gerlach, and C. E. Young (2007). “An empirical analysis of open source software developers’ motivations and continuance intentions”. In: *Information and Management* 44.3, pp. 253–262.

Xu, B., D. R. Jones, and B. Shao (2009). “Volunteers’ involvement in online community based software development”. In: *Information and Management* 46.3, pp. 151–158.

Yin, R. (2003). *Case Study Research - Design and Methods*. 4th. Applied Social Research Methods. Sage.

Yourdon, E. (1977). “The choice of new software development methodologies for software development projects”. In: *the June 13-16, 1977, national computer conference*. New York, New York, USA: ACM Press, pp. 261–265.

— (1998). “A tale of two futures”. In: *IEEE Software* 15.1, pp. 23–47.

Zittrain, J. (2009). *The Future of the Internet: And How to Stop It*. Penguin.

Zizek, S. (2012). *Organs without Bodies: On Deleuze and Consequences*. Routledge Classics.

Zourabichvili, F. (2006). “Kant avec Masoch”. In: *Multitudes* 25.2, p. 87.

— (2012). *Deleuze: a Philosophy of the Event*. Ed. by G. Lambert and D. W. Smith. Plateaus: New Directions in Deleuze Studies. Edinburgh University Press.

Zuboff, S. (1989). *In The Age Of The Smart Machine: The Future of Work and Power*. Basic Books.

Glossary

ADL Archetype Definition Language. A language used to define the constraints set on archetypes.

AM Archetype Model. The layer which constrains the reference model and builds complex clinical requirements, but not yet instantiated to specific contexts.

AQL Archetype Query Language. A language used to retrieve clinical data from archetypes.

CC-BY Creative Commons license with a requirement for attribution of source work to original license author.

CC-BY-SA Creative Commons license with a requirement for attribution of source work to original license author. Derivative works must also be licensed as CC-BY-SA.

CKM Clinical Knowledge Manager. openEHR's repository where archetypes are stored, analysed and discussed.

HL7 HL7 International is a not-for-profit organisation that is, among other things, concerned with EHRs and the detailed clinical modelling.

IHTSDO A not-for profit association with close ties to openEHR (the two foundations almost merged). Owners of SNOMED CT.

RM Reference model. The basic classes which act as building blocks for the creation of more complicated clinical requirements.

SNOMED A terminology project owned by IHTSDO which aims to define low-level, biological concepts.

Templates Often the result of further constraints and amalgamation of archetypes. Refers to specific use contexts.

Appendices

Appendix A

Pilot Protocol

Abstract

This protocol introduces the subject matter for a PhD research study concerning requirements in open source and their link to IT innovation. OpenEHR is an interesting case to consider in this context given the complexity of health care informatics. The study aims at deconstructing narrow meanings of what requirements are and to offer a better understanding of requirements and their processes in open source.

Two principal ways of collecting data are proposed : retrospective and prospective exploration of requirements. Further, it is hoped that interviews will provide additional contextual information to the requirements elicited.

Introduction - Aims and Objectives

Why do I want to study OpenEHR? To answer this question, we must first introduce the subject matter of my PhD.

The research is concerned with the study of requirements. These can be understood as embodying the vision of various stakeholders for a system-to-be. They usually are understood to represent an attempt to model a future vision around a specific technical artefact. Both the requirements and the process in which they are embedded have a fundamental relation to IT innovation.

My research is based on questioning the narrow view of requirements as being only specifications for the technical elements of a system-to-be. The concepts of emergent naratives, improvisation, control, and hypomnemata are used to deconstruct such a common understanding of requirements, and the various systems in which they are embedded.

Open source provides an interesting environment in which to study requirements given how open source development is often portrayed as a negotiated, informal and collaborative development paradigm where participants speak directly of their needs and 'itches'. Looking beyond the commonly researched open source projects, we plan to focus on projects in vertical markets, where requirements can be vague, ambiguous and uncertain and where the know-what is arguably more important than the know-how. The implications of hidden complexity in the process of creating new software in such domains might be seen in the existence of new or specific coordination mechanisms to control and harness open creativity within the dynamics of the specific communitiesÕ style of collaboration.

Following the principal bodies of literature concerning open source requirements (Mockus et al. 2002; Noll and Liu 2010; Scacchi 2002; Scacchi et al. 2006), we pose several hypotheses to validate, refute and refine:

H0 that OS requirements are ad-hoc and informal;

H1 that the OS requirements processes are minimal and varied;

H2 that IT innovation in OS is user-led or community-led through the medium of requirements;

H3 that OS demands a different requirements process;

H4 that this process has a formative role on requirements

OpenEHR is distinct given the sector it is in and it seems to provide an excellent setting to study requirements and their role in the process. Its distinctive nature makes it an interesting case to study in its own right. The project deals with complex expert domains potentially affecting stakeholders of varied expertise. The context of healthcare is also strictly regulated and constrained by laws and professional rules, and the project has important strategic goals which need to be satisfied to a high degree (including compliance against legislation and standards).

OpenEHR is interesting also for its attempt to define archetypes, specifications which define and express business or information requirements in their own right.

Method

Why a case study?

A case study is an appropriate way to conduct this research for the following reasons:

1. The context is important to the subject
2. It follows that not only “objective” artefacts are useful, but also interpretations of their meaning or purpose in understanding the artefact and its place in the world
3. The case is unique and revelatory (Yin 2003)
4. The research needs both primary and secondary evidence
5. Requirements are often embedded with not only tacit, but also difficult to express or unknown (Alvarez 2002). These, one can surmise, can only be identified through interviews

What data to collect?

The core of the subject of study is requirements and their relation to innovation, open source development and work practices. So, what to collect? Requirements are the ‘objects’, ‘meanings’, ‘concepts’ to be observed and from which stem questions about their life-cycles and what and how they influence and are influenced by. The other revolving issues, or aspects of them (e.g. innovation), it is hoped, will be revealed through the observation of how requirements are handled.

Therefore, requirements represent the primary data that we wish to study in this research. Still, requirements themselves are embedded in a process, and it can be hard to separate one from the other. This increases the complexity of observing requirements but also reinforces the need to provide interpretations as to their purpose and influence.

What is a requirement?

Despite the narrow, software engineering definition introduced in Section A, requirements can take various subjective or constructed forms. Should goals, desires, objectives, user needs be considered as valid data requirements (e.g. straying beyond the functional)? How are they expressed? And how and where should we look for them? Do they all have the same appearance? Should they all be treated in the same way? Are they goals, objectives, functional and non-functional requirements can have different definitions and purposes?

The question of what is a requirement is a difficult one. From a narrow point of view, a requirement is a technical specification for a system-to-be. But requirements can be much more than that and do not necessarily have a physical representation (it could be say, an idea, or a set of tacit understandings) This is, in effect, one of the intended contributions of this research: to depart from a narrow perspective of requirements. The work thus attempts to interpret the meaning of a requirement and its implications. Nonetheless, as a starting point, and in order to derive possible interpretations of requirements, the initial focus is on the written artefact, in its physical or digital form. From there, we can trace its role, significance and legitimacy which will be researched.

The project scope thus helps manage the difficulty of multiple requirement forms and expressions in the real world of a case study. By starting with the narrow, software definition of requirements, and trying to discover what other meanings they take throughout development, then it will be possible to discover other interpretations of what requirements are or can be. We hope to start the investigation from what OpenEHR project considers as a requirement, hence reducing the initial definition of a requirement to a discrete, identifiable set.

How will data be collected?

Algorithmically, data will be collected in the following way:

- Collect a set of requirements from one or more recent project documents
- Data is collected through interviews and observation of workshops in two different ways depending on the state in which the “in use” requirement is (inception, formally described)
  - Retrospective exploration where requirements are traced back in time and understand how the requirement was elicited, formed or discussed
  - Prospective exploration where requirements are followed forwards in time

These two data collection methods each have their pros and cons. The retrospective, principally, assumes that existing records will provide sufficient information and context and will not be simple rationalisations of what actually took place. Interviews could help solve part of the problem but are also subject to post-rationalisations. The context, thus, might be simplified or biased when using only retrospective archival exploration.

Prospective exploration, on the other hand, allow for experiencing the events ‘live’. This allows for a realistic observation of the context and developments around the life of the requirement. On the other hand, there might be a lack of online documentation to base observation upon, requiring the use of interviews or participant observation. Consequently, this method is more costly in terms of time and resources.

Both methods are complementary. By using both retrospective and prospective data collection methods, it is possible to paint a more accurate representation of the events taking place in shaping requirements or how requirements themselves shape their context. For example, finding evidence of post-rationalisation could help understanding some aspects of the work revolving around requirements, with say, the need of formalisation to teach new comers how requirements are handled.

This project therefore expects to deal with multiple sources of evidence: semi-structured interviews, archival analysis and participant observation (ie. discussions in forums, seminars, workshops or conferences). Open source research, drawing on its philosophy of opening its archives tends to mix archival exploration and participant observation. In addition, attendance to key conferences can provide an added source of evidence which might reveal, other, more intricate ways as to how requirements are shaped.

Data collection considerations

It is hoped to interview various types of contributors in a multiple round scheme, where the feedback of the first round would be used in the second round:

- High-level staff connected to one of the decision making bodies, principally regarding strategic decision making
- Geographically distributed, "normal" contributors with less formal connection to decision making boards
- From an understanding of the requirements workflow, interview stakeholders from which expert knowledge has been extracted such as clinicians and patients
- Possibility to become a potentially active or passive participant, avoiding, if possible, becoming a lurker

This research follows several guiding principles:

- Privacy will be guaranteed. Interviewees will be asked whether they agree to take part and if they agree to have their input published with anonymised aliases
- Interviewees will be asked whether they wish to carry on with the interview and will be allowed to stop whenever they choose.
- The interview sessions will be audio recorded only if the interviewee expressively accepts
- Data published on the OpenEHR website will be considered as 'public' and will be used unmodified in accordance to the applicable license

How will evidence be triangulated

Case studies often need to establish construct validity and reliability (Yin 2003). Beyond the simple need of showing the methodological solidity, the rational for having multiple sources of evidence is justified for this research since it tries to look at how requirements are interpreted and used. It is thus important to go beyond the interpretations of the researcher or the "primary" stakeholders and seek explanations from people involved in other ways. In this way, a more accurate reality, or realities might be constructed.

How will evidence be analysed?

We plan to analyse data following two theoretical concepts: one is hypomnemata, and the other is improvisation. The study thus tends towards a phenomenological account of the everyday dealings surrounding requirements. These two principal concepts offer an interesting tension: one presupposes order, control and rational and the other freedom, creativity and chaos. It is a small step to call one a closed concept, and the other an opened one.

Open source advocates openness. This usually means that the code itself is open to inspection, but tends to apply to other practices and artefacts such as rules and methodologies. It is more than a legal matter and is often seen as a cultural or philosophical principal. It is, however, imaginable that under certain circumstances the development methodology has to impose a certain order or control. This might especially be the case in health systems for the reasons outlined in Section A (compliance to laws, standards, safety and critical properties). The point is to look at successful or deemed to be, mission critical systems and understand the possible need for formality, and how it might, say, help or induce creativity. This control might not be obvious and could, for example, be embedded in the narrative of a requirement, and the way debate evolves.

The research does not deal with good or bad or offer such judgments in this respect. It simply supposes that the requirements process will necessitate some elements of control, especially in distributed collaborative domains, and more to the point in safety relevant systems.

What are possible interview questions?

The following is a selection of possible questions to give a flavour of the work:

- How are requirements elicited, sought out or captured by OpenEHR, and through which means?
- How are requirements prioritised (through board reviews it seems)?
- How does a requirements form? How is it in turn transformed? Into what?
- How do requirements serve as part of validation or testing?
- How is the community involved in the definition of requirements (and which community)?
- How are stakeholders defined in OpenEHR?
- Is the context of use analysed to “find” requirements?
- How do the licensing issues influence archetypes/requirements definition?
- Are there internal documents not publically available?

Potential Timeline

In terms of planning, this research enters its second year in which we expect to collect data, leaving the third year to writing up. Given the scope of a PhD research, we plan to collect data for half a year, until June approximately.

Appendix B

Code

Listing B.1: Caption example

```
# Globals
name = ""
year = 0

def main():
    printWell(count(main()))

def main(year):
    print ("Clinical␣"+year)
    printWell(count(readSubject(year,"-clinical")))
    print ("Technical␣"+year)
    printWell(count(readSubject(year,"")))
    pass

# Two for loops, one to cycle through files, the other does the actual parsing.
# Splits file into lines and lines into words. Makes a basic check that the line
# being read is actually the subject, and not a response email (with >>).
def readSubject(year,list):
    months = ["January", "February", "March", "April", "May", "June", "July",
            "August", "September", "October", "November", "December"]
    file = fileinput.input()
    subject_list = []
```

```
  for x in months:
    file = open(str(year)+"-"+x+list+".txt")
    for line in file:
      splitted = line.split()
      if "Subject:" in splitted:
        if splitted[0] != "Subject:":
          pass
        else:
          subject_list.append(line)
  return subject_list

# Counts occurrences of same elements from a list, returns an object of class Counter
def count(list):
  occurrence = Counter()
  for line in list:
    occurrence[line] +=1
  return occurrence

# Prints each element of the Counter object as a line by joining the newline string (\n)
def printWell(counter):
  print ("\n".join(str(element_and_count)
  for element_and_count in counter.most_common(15)))

def printCounterYear(counter,year):
  pass

if __name__ == '__main__':
  main(sys.argv[1])
```

Appendix C

Selected Movements in openEHR Lists

Table C.1: Selected movements in the 2009 openEHR technical list

Movement	Summary	Transformation	Time
'Why is the editor not opening ADL files?' (Demographics)	A question on the reasons why the archetype editor could not open demographics archetype files unpacks many other questions. The importance of demographics archetype opens the door to a discussion on requirements processes, control, open source. The predominant role and intentions of Ocean—one of the principal contributors—is put into question. The basic, original purpose of openEHR is discussed and used as leverage	Reqs -> ReqProc, Control, OS, openEHR goals	Mar, Apr 2009

'License and copyright of archetypes'	Discussion started from a participant question wondering about the legal implications from openEHR maintaining a copyright notice in the CKM. The ambiguity of the license used disentangles the status quo and forces participants to rethink priorities. The variety of licence choices puts requirements and their processes in the forefront, because they define who, how and what changes are possible to the core openEHR deliverables: the community developed archetypes. Requirements finally ask: what does it mean to be open source?	Val -> Reqs, ReqProc	Sep, Oct 2009
'openEHR community on Google Wave'	The use of Google Wave as an alternative collaboration and requirements process initiates a discussion on the accessibility and searchability of openEHR, its CKM archetype repository, and the wider discussions on requirements to a wider audience. The role and ownership of the CKM is discussed and Wave proposed as a non vender lock-in alternative. Comparisons with other open source projects are made and put in light of openEHR's requirements and processes	ReqProc -> OS -> Reqs	Nov, Dec 2009
'Wisdom of the Crowds'	Another thread on the role of openEHR and open source in sourcing and articulating requirements	OS -> ReqProc, Reqs, Val	Feb 2009

'Layers of interoperability, OWL and openEHR'	Both technical and high-level philosophical discussion about the role of formalisms in the creation of interoperability layers in openEHR and in general. How, and is it possible at all, to achieve complete interoperability? The discussion centers on the requirements for systems to be interoperable	Reqs -> Reqs, ReqProc, Terr	Apr 2009
'Issues around UI technologies and bindings to back end'	Technical discussion on the integration of UI technologies and their representation of archetypes. Discussed are requirements qualities of complexity, ubiquity, and interoperability, as well as values of aesthetics and technical criteria such as use of the Model-View-Controller paradigm. The nature, role and purpose of archetype requirements are weighed along with the responsibility they cede to templates to present the information they hold. Tools are shown to be an important aspect of showcasing potentiality for expansion and evolution	Req -> Req, Terr, Val	Jul 2009
'Document desparation'	Desperation of a participant to what he considers a lack of coherence to the goals of openEHR and open source to have non open source programs in the production pipeline. Not only does it conflict with the ideological foundation, but also hampers collaboration and uptake of openEHR by creating an forced, untrustworthy, incoherent process of participation	OS, Val -> Id, ReqProc	Sep 2009

'RM Versions'	Initial thread on importance for interoperability of declaring and validating archetypes against versions of reference models. The discussion becomes a short debate over the meaning of interoperability in EHRs and how uncertainty and newly acquired experience can incite breaking changes across reference models ' "Future-proof" at risk! was: RM Versions'	Reqs -> Terr, ReqProc	Feb 2009
'CQuantityItem units not empty'	The requirement of a specification is not clear and gives rise to an unexpected discussion on what units an object of class CQuantityItem should hold	Req -> Terr -> Req	Jul 2009
Historicity and rigour	Sometimes used together, historicity, on its own, refers to the use of existing material which is shared for a purpose. Many times, this purpose is to give rigour to a requirement by providing proof of its necessity	Hist -> Req Hist -> Rigour -> Req Rigour -> Hist -> Req	Jan, Feb, Mar 2009

Table C.2: Selected movements in the 2010 openEHR technical list

Movement	Summary	Transformation	Time
'Interoperability with HL7'	A question on the interoperability between openEHR and HL7 launches a discussion on the merits of the openEHR approach and its relation to competing philosophies of interoperability in healthcare. The discussion casually and continuously links specific, detailed requirements (e.g. attribute number) to high-level ideas of complexity and requirements processes. The discussion reveals what it means to be interoperable for openEHR and relates to the world as-is. Interoperability is a rare controversy which spilled on these other threads: 'ISO 21090 data types too complex?'. It has maintained its controversial identity throughout these threads, showing the importance of this requirement and the difficulty to arrive at a negotiated understanding of the necessities it requires to be satisfied	Reqs -> Reqs, ReqsProc, Strat, Id	Jan, Feb 2010
'GUI-directives/hints again (Was: Developing usable GUIs)'	Started as a guidance request for UIs and ended mostly discussing the space of UI requirements in the modelling of EHRs and the role of templates and archetypes	Reqs -> Req, ReqProc, Terr, Val	Dec 2010

'ISO 21090 data types too complex?' and 'More on ISO 21090 complexity'. Spills over to the 'HL7 modelling approach' thread in November 2010 and 'World Peace'	Large discussion started on the complexity of an ISO standard defining data types. The thread ends up discussing multiple themes: territories that affect requirements sourcing, how inappropriate requirements process influence the territories explored, and the values behind the requirements processes, which is a latent concern in openEHR. This thread also indirectly reveals the majoritarian concern of openEHR.	Terr -> Reqs (early) Terr -> ReqProc -> Terr (mid) Val -> ReqProc (concern)	Nov 2010
'HL7 modelling approach'	Spill from thread 'ISO 21090 data types too complex?'. Discusses differences of approach between HL7 and openEHR, where one follows a design-by-committee, and the other, as argue a core openEHR participant, a more meritocratic ideal	ReqProc -> Terr	Nov 2010
'Why is OpenEHR adoption so slow?'	Spilled from 'Articles on Healthcare, Complexity, Change, Process, IT and the role of openEHR etc' thread. Discusses lack of international representation of openEHR and how standards and requirements should be developed. Debates the role of certain actors in adoption of requirements as standards	Terr -> ReqProc	Oct, Nov 2010
'Term bindings in archetypes and templates'	Detailed technical discussion on the appropriate space for the binding of terms to archetypes and templates. Discusses the role, purpose and responsibilities of clinical semantics, and the representative potential of clinical concepts and its rapid evolution	Reqs -> Terr	Mar, Apr 2010

‘Run-time name constraints and appropriate use of terminologies’	Technical discussion on name constraints and whether these should be defined at archetype or at term level	Reqs -> Reqs, Val	Aug 2010
‘openEHR-13606 harmonization CR regarding CLUSTER/TABLE etc and ENTRY/OBSERVATION’	Descendent of ‘ISO 21090 data types too complex?’. A technical thread on a particular change request. Discusses concerns found in other threads such as requirements qualities (e.g. simplicity and complexity), but also what is the right standardisation process	Terr -> Req, ReqProc Req -> Req (concern)	Nov 2010

Table C.3: Selected movements in the 2009 openEHR clinical list

Movement	Summary	Transformation	Time
‘License and copyright of archetypes’ and ‘[openEHR-announce] Interim Statement on Copyright and Licensing of Archetypes’	Discussion mirroring the one by the same name in the technical list. Follows up an announcement email of the consideration by the board of the license choices	Reqs -> ReqProc, Terr, Reqs, Val	Sep, Oct 2009
‘Cancer treatment archetypes’	Interest in developing archetype for research purposes. Showcases the emergence of requirements from unexpected actors	Id, Terr -> Req	Nov 2009

‘openEHR community on Google Wave’	Mostly people asking for Google Wave invites. Started as discussion about involving Google Wave as collaboration tool. Some participants complain about CKM not being openly searchable. Mirrors the discussions on technical list	ReqProc -> Id, Val	Nov, Dec 2009
‘informal poll: openEHR conference’	Poll about where to hold next openEHR conference	Transformation	Nov, Dec 2009
‘Adverse Reaction archetype – review round initiated’	Thread asking for participation in a review round for the ‘Adverse Reaction’ archetype	Req -> ReqProc, Req	Jul 2009
‘Modeling reference ranges’	A question about a requirement’s functionality shows the necessity to follow requirement rational further along	Req -> Req, Val, Id	Oct 2009

Table C.4: Selected movements in the 2010 openEHR clinical list

Movement	Summary	Transformation	Time
‘Why is OpenEHR adoption so slow?’	Mirrors same discussion from technical list, but steers into new direction debating need to showcase openEHR relative to competitors. ‘Interoperability’ as a requirement is discussed. Influences prioritising requirements and business model, and how to engage with openEHR	ReqProc, Terr -> ReqProc, Terr, Reqs	Nov 2010
‘Decision Support Providers’	Mirrored in technical list. Reaction to a blog post on standards and decision support systems, and its relation to openEHR. Little relevance in terms of requirements.	N/A	Jun, Jul 2010

IHTSDO meeting - term binding presentation available	Mirrors technical list. Coordination thread and information of presentation by a participant. Indicative of openEHR's necessity to seek alliances	Id -> Terr?	May 2010
??? DV_CODED QUANTITY	Question about modelling in openEHR a scale which mixes symbols, integers and real numbers. Requirement emerged from a participant's work. The discussion tries to integrate the new requirement while protecting the core requirements' integrity (e.g. the meaning of archetypes and data values)	Req -> Req, Val	Apr 2010
About parallel archetype editing	Discussion started by a possible implementer concerned about the specialisation of regional archetypes and whether they could work on the same archetype at the same time. Gives rise to a small discussion about the convention of identifying specialised requirements	ReqProc -> ReqProc, Reqs	Jul 2010
GUI-directives/hints again (Was: Developing usable GUIs)	Mirrors technical list. On the clinical list, the first thread is titled 'New requirements from endoscopy (was Re: GUI-directives/hints again (Was: Developing usable GUIs))'. Discusses 'purity' of requirements and disruptive change	Terr -> Reqs	Dec 2010
Comment on openEHR-EHR-CLUSTER.inspection-skin-wound.v1 archetype	Small discussion on the particularity of wound animals in the wound archetypes	Reqs, Terr -> Reqs, ReqProc	Feb 2010