Integrating Unlocking of Organizational Paths into an Agent-Based Simulation Model

Inaugural-Dissertation

zur Erlangung des akademischen Grades eines Doktors der Wirtschaftswissenschaft des Fachbereichs Wirtschaftswissenschaft der Freien Universität Berlin

vorgelegt von: Dipl. Ing. Felix Obschonka

aus: Filderstadt

Berlin, 2014

Erstgutachterin: Prof. Dr. Natalia Kliewer

Zweitgutachter: Prof. Dr. Oliver Baumann

Tag der Disputation: 27. Mai 2014

Danksagung

An dieser Stelle möchte ich mich bei den Personen bedanken, welche mich durch Ihre Unterstützung und Ihr Mitwirken bei der Fertigstellung der Dissertation unterstützt haben.

Zu allererst gilt der Dank meinen Betreuern, welche stets ein offenes Ohr für mich hatten und mich neben fachlichen Ratschlägen in schweren Phasen unterstützen. Im Besonderen danke ich Natalia Kliewer für Anregungen bezüglich der Modellierung und der technischen Implementation, Oliver Baumann für seine Geduld, seinen hohen Einsatz, wertvollen Tipps zur Modellierung und die Zeit in Dänemark, sowie Arne Petermann dafür, dass Theorie und Modell miteinander harmonieren und den unzähligen Stunden die er für mich und die Arbeit opferte.

Neben dem Betreuerteam danke ich Georg Schreyögg für die Unterstützung und Ausbildung durch das Graduiertenkolleg sowie der Deutschen Forschungsgemeinschaft für das großzügige Stipendium und die finanzielle Unterstützung der Auslandsaufenthalte. Nicht zu vergessen sind James March von der Stanford Universität, welcher mir mit Modell und Rat zur Seite stand, die Mitglieder der Strategic Organizational Design Unit an der University of Southern Denmark, welche sich die Zeit nahmen mein Vorhaben zu besprechen und wertvollen Input lieferten, Sendil Ethiraj von der London Business School, welcher mich in der Forschung unterstützte und den Aufenthalt in England bereicherte, den Kollegiaten, allen voran meinen Büronachbarn Anne Petscha, Benjamin Schulte und Sibel Siray, sowie Christian Dohnert, Florian Otto und Martin Kasparick, welche mir bei Fragen rund um die Programmierung der Simulation geholfen haben.

Besonderer Dank gebührt zuletzt meinen Eltern Frank und Monika und meiner Schwester Nadine, welche stets an mich glaubten und in den letzten Jahren viel zurückstecken mussten aufgrund arbeitsamer Nächte und Wochenenden.

Abstract

While models of path dependence can explain how organizations fail, the notion of organizations overcoming lock-in remains ambiguous. Although first means to unlock paths were proposed in management literature, systematic testing if and how unlocking is possible still needed further elaboration. Therefore the dissertation at hand sheds some light on unlocking by integrating the logic of escaping organizational paths into a four-phase model of path dependence and testing how turnover, reconfiguration of the organizational structure and a top management team affect the process of unlocking organizational paths. These means were selected because prior literature hinted to their effectiveness in unlocking paths. To show how unlocking can occur, a computer simulation model based on the organizational learning model of March (1991) is derived, that takes into account individual learning based on the similarity between agents.

The results of computer experiments show that heterogeneity can be preserved over the path formation process under certain conditions. Furthermore, the findings emphasize that heterogeneity in conjunction with an exogenous shock facilitates the unlocking of organizational paths according to the four-phase model of path dependence. Without such exogenous pressure the organization is, in contrast to prior models, not able to escape lock-in.

With concern to the question how turnover, reconfiguration and a top management team influence an organizations ability to unlock paths, the results vary depending on the similarity parameter. The similarity parameter takes into account that organizational members prefer similar individuals to learn from and avoid to learn from dissimilar individuals. If the similarity value is set to a high value, learning from dissimilar individuals is impeded, set to low values individuals are more likely to learn from dissimilar others. The computer experiments indicate that for low similarity values turnover proves to be more effective compared to reconfiguration, and for high similarity values, reconfiguration of the organization through rotation of agents is advisable. In contrast, replacing the top management team proves to be effective over all similarity values. These differences in the effectiveness of

means can be attributed to the presence of heterogeneity in individual beliefs, their capability of learning from others and the coordination between individuals in the organization.

Zusammenfassung

Das Berliner Drei-Phasen-Modell pfadabhängiger Prozesse erklärt, wie die Handlungsfähigkeit von Organisationen aufgrund selbstverstärkender Effekte eingeschränkt werden kann. Das Phänomen Pfadbruch, bei welchem Organisationen etablierte Pfade verlassen, wird darin nur unzureichend beschrieben. Obwohl in der Managementliteratur verschiedene Möglichkeiten zum Aufbrechen von Pfadabhängigkeiten diskutiert werden, fehlt eine systematische Überprüfung, ob und wie organisationale Pfade gebrochen werden können. Die vorliegende Arbeit beleuchtet diesen Aspekt durch die Integration der Logik des Pfadbruchs in das Vier-Phasen-Modell von Sydow et al. (2005). Anschließend wird getestet, wie Fluktuation, Restrukturierung und ein Management-Team einen Pfadbruch beeinflussen. Um diesen nachzuweisen, wird eine Computersimulation verwendet, welche auf dem organisatorischen Lernmodell von James March (1991) basiert. Unter der Berücksichtigung, dass Individuen in ihrer Fähigkeit zu lernen limitiert sind, wird ein *similarity* Parameter eingeführt, der die Lernwahrscheinlichkeit, und damit den Lernerfolg, an die Ähnlichkeit von Individuen innerhalb einer Organisation koppelt. Ein hoher *similarity* Parameter erschwert das Lernen von Individuen mit geringer Ähnlichkeit, während ein niedriger *similarity* Parameter Lernen von Individuen mit geringer Ähnlichkeit erleichtert.

Die Ergebnisse der Computerexperimente zeigen, dass unter bestimmten Gegebenheiten Heterogenität während des Pfadformierungsprozesses erhalten bleiben. Verbunden mit einem exogenen Schock wird durch Heterogenität das Aufbrechen von Pfadabhängigkeiten ermöglicht. Dies bestätigt ein Vier-Phasen-Modell pfadabhängiger Prozesse, in welchem der Pfad nach der Lock-In-Phase gebrochen wird. In Übereinstimmung mit der Pfadtheorie können Pfade dabei nur durch Druck von außen verlassen werden.

Inwiefern Fluktuation, Restrukturierung und ein Management-Team die Fähigkeit einer Organisation Pfade zu verlassen beeinflusst, variieren die Ergebnisse. Die Experimente zeigen, dass für niedrige Werte des *similarity*-Parameters, das

Auswechseln von Individuen effektiv ist, während für hohe Werte des *similarity*-Parameters, Restrukturierung durch Rotieren von Individuen innerhalb einer Organisation empfehlenswert ist. Ein Auswechseln des Management-Teams manifestiert sich über alle *similarity* Werte hinweg als effektiv. Diese Unterschiede in der Effektivität der Strategien zum Pfadbruch kann durch die Präsenz von Heterogenität in den Überzeugungen von Individuen, durch die Fähigkeit von anderen Individuen zu lernen und durch die Koordination zwischen Individuen erklärt werden.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CD	Compact Disc
CEO	Chief Executive Officer
CFC	Chlorofluorocarbon
COO	Chief Operating Officer
DAT	Digital Audio Tape
DVD	Digital Versatile Disc
HFC	Hydrofluorocarbons
IBM	International Business Machines
R&D	Research and Development
TMT	Top Management Team
VCR	Video Cassette Recorder
VHS	Video Home System

"Each problem that I solved became a rule which served afterwards to solve other problems."

- Rene Descartes (1596-1650)

1. Introduction

Turbulent environments demand organizations to balance flexibility and efficiency in order to ensure competitiveness and long-term success (Bingham, et al., 2007). Especially in fast moving and hypercompetitive industries, strategic flexibility and steady rejuvenation are vital for organizations in order to respond to changes in their environment (Volberda, 1996). An example for a company that managed to remain flexible and adapt to changes is International Business Machines (IBM). During its history, IBM faced several crisis, due to changing market landscapes, and underwent successful organizational transformation processes (Maney, 2003). While there are organizations, like IBM, that master profound transformation processes, numerous cases of organizations failing to adapt to new realities are known as well (Romanelli & Tushman, 1994). For example, Kodak and Polaroid did not achieve to change its business model from analog to digital imaging technologies (Lucas & Goh, 2009; Tripsas & Gavetti, 2000). Eventually, the once successful companies had to file for bankruptcy. Because of such detrimental effects, management scholars are interested in barriers that impede organizational change processes (Tidd & Bessant, 2009). Among the large numbers of explanatory approaches, the concept of organizational path dependence gained momentum in management research (Sydow, et al., 2009). Initially, the concept was utilized in the field of economics to explain suboptimal market outcomes, due to the diffusion of inefficient technologies (Arthur, 1989; David, 1985). Management scholars adopted and transferred the concept to the domain of organizations in order to explain how the scope of strategic choices narrows down over time until strategic flexibility is lost (Holtmann, 2008). Over the last decade, most of the research covering organizational path dependence focused on this issue, revealing how once successful firms were sucked into the dynamics of self-reinforcement, eventually ending up in a suboptimal stable equilibrium (Ericson & Lundin, 2013). Having an understanding of the effects of self-reinforcing mechanisms on the formation of organizational paths, as well as contextual circumstances responsible for paths to emerge, is useful to prevent harmful behavior in the first place. But while the concept of path dependence can explain why and how Polaroid (Tripsas & Gavetti, 2000) or Kodak (Pandza & Thorpe, 2009) failed, it still misses to explain cases like IBM, where organizations are stuck, but eventually change to adapt (Gerstner, 2002). A possible barrier studying how organizations break free from paths is that the concept of path dependence originally did not intend to explain unlocking. Although means were proposed on how to unlock paths (Castaldi & Dosi, 2005; Sydow, et al., 2005), prior literature remains purely theoretical and does not prove if these means may actually break paths. Apparently, since then, only little effort has been put into extending the theory to include the logic of unlocking paths into the stage model of organizational path dependence proposed by Sydow, et al. (2009) (for an exception see Ericson and Lundin (2013)). But, knowing the potentially self-destructive consequences of lock-in, integration into the framework to examine the conditions and the process of unlocking is needed. The dissertation at hand therefore attempts to shed light on the field of unlocking organizational paths by integrating the logic of unlocking into the organizational model of path dependence. This is achieved through merging the original model of technology adoption with the organizational three-phase framework of path dependence, and extending it with mechanisms and intentional means that allow for unlocking. Therefore, the research contributes to organizational theory, and provides a framework to explore new directions in path research. Besides that, the work aims to stimulate the emergent discussion on path *in*dependence (Ericson & Lundin, 2013). That is, scholars should examine ways of how organizations can prevent, control, escape, or forecast rigidifying processes in organizations.

1.1 Research Objective

Objective of the work is to integrate the logic of unlocking into the theoretical three-phase model of path dependence. Furthermore, means for unlocking paths by intentional actions through the management of an organization are derived. To achieve this goal, the task is to develop a model left ajar on the original definition of path dependence, as a stochastic process governed by contingency and self-reinforcement, while simultaneously considering the possibility of unlocking. Building upon prior literature in organizational path research, the case of individuals unlocking paths in organizations through their behavior has to be viewed in particular. After the model that takes unlocking into account has been developed, a simulation model which tests the effects of the individual behavior on unlocking has to be build. Furthermore, intentional actions of the management to break paths have to be derived from literature and included in the model, and must be tested subsequently. Literature on organizational change frequently names reconfiguration of the organization (Karim, 2006; Karim & Mitchell, 2000), labor turnover (Nystrom & Starbuck, 1984), and top management influence (Wiersema & Bantel, 1993) as means to overcome rigidities. These organizational means to unlock paths are tested with the help of a computer simulation model.

1.2 Research Questions

Based on the previous remarks, first the issue on how to extend the concept of path dependence has to be answered and second, intentional means influencing the breaking of organizational paths must be explored. Therefore, two independent research questions arise:

Research Question 1:
How can the logic of unlocking be included into a model of organizational path dependence?

Research Question 2:
How do (a) turnover, (b) reconfiguration, and (c) a top management team affect the probability of unlocking organizational paths?

1.3 Dissertation Outline

The dissertation comprises seven chapters to answer the research questions. This chapter briefly referred to the motivation for conducting research on the unlocking of organizational paths, and stated the research questions. In order to obtain a deeper understanding of the path dependence concept, the current state of path research is reviewed in the second chapter. Commencing with increasing returns in technology adoption and the constituting properties of path dependence, the concept is subsequently transferred to an organizational context. Here, a theoretical framework provides the basis for studying organizational paths and gives guidance on how to capture path dependence in organizations. Having an understanding of how paths evolve, means for unlocking organizational paths are derived in a next step. In order to close the research gap by answering the research questions, a scientific methodology has to be found. Chapter three hence argues that the computer simulation is an appropriate method for examining organizational paths. As simulation studies are scarcely used in management research, characteristics and relevance of the method are briefly explained. Finally, chapter three introduces a structured guide for conducting computer simulations as a fundament for theory building and the dissertation at hand. Referring to the guide, the fourth chapter discusses established simulation models in management and path research. This facilitates the selection of a reference model, incorporating the properties of path dependent processes. After implementing and validating the reference model, it is gradually extended to include independent and dependent variables of the research hypothesis. The underlying assumptions of the model and the parameter implementation are comprehensibly stated to convince the reader of the model. In chapter five, the extensive simulation framework is used to conduct three experimental studies for investigating how organizations may unlock paths. The first set of experiments focuses on the emergence of organizational paths, dependent on the micro behavior and the effect of exogenous shocks on the ability to unlock paths. The second experimental study analyzes how restructuring, through the rotation of individuals within the organization, and employee turnover affect the breaking of organizational paths. In the last set of experiments a hierarchical level, having normative influence on individuals in the organization, is introduced. Then, it is examined how this may contribute to the unlocking of organizational paths. The results of the experiments are summarized in chapter seven. Furthermore, chapter seven provides an outlook for further research and highlights the limitations of the dissertation.

2. Theoretical Background

The broad application, and often imprecise definition, of “path dependence” in social research makes it necessary to state what is actually meant by the term *path dependence* (Sydow, et al., 2009). A precise definition of path dependence will help to understand the difficulties when it comes to unlocking. Furthermore, the research must be embedded into a broader understanding of how organizations behave in general. Therefore, this section will provide a literature review on the concept of path dependence, give a clear definition of path dependence, and link it to prior organizational theory. Finally, the literature on unlocking of paths is reviewed and a four-stage model of path dependence is proposed.

2.1 Path Dependence in Technology Adoption

Increasing returns, leading through contingent "small events" to inefficient market monopolies, are central to the concept of path dependence in technology adoption (Arthur, 1989; David, 1985).[1] Considerations on how increasing returns affect economic outcomes have a long history, and date back to noble-prize winning research on international trade and monopolistic competition (Dixit & Stiglitz, 1977; Krugman, 1979; Marshall, 1890).[2] But, it was only in the 1980s that the potentially negative effects of random small events became part of the discussion on increasing returns in economics (Arthur, 1983; David, 1985). Later, Arthur expressed the idea of how increasing returns in technology adoption may lead to a suboptimal stable equilibrium state through contingent small events, and named the resulting process “path dependent” (Arthur, 1989). Research on increasing returns was already motivated by the discussion on circumstances of how an inferior solution could achieve market monopoly (Krugman, 1979). But, Arthur in

[1] Small events are “*outside our knowledge”* (Arthur, 1994: 14), taking place at the beginning of the process, and not averaged away over time. Sydow, et al. (2009) draw an analogy to the butterfly effect. The butterfly effect explains how small changes in initial conditions may have a large impact on the whole system.

[2] See Krugman (1998) for a detailed research chronology on increasing returns in the field of business and economics.

particular considered the case of how *contingent events* under a regime of increasing returns may result in locking an economy to an inferior technology (Arthur, 1989). The main argument for this was that if economic agents have to choose between different technologies for adoption, chance could give a competitive edge for one technology at the very beginning of the diffusion process. As a result of the early lead, the utility of adopting this technology increases and, because of that, the technology is further adopted by utility maximizing actors.[3] A self-reinforcing circle of increasing utility accompanying further adoption may eventually lead to a situation, where one technology crowds out alternative technologies and dominates the market. Strikingly, the development of which technology is cornering the market solely depends on the self-reinforcement of initial random small events, and not on the long run performance of the technology. The explosiveness of this argument lies in the statement that historical contingencies in the adoption process have serious effects on the final outcome, and may even lead to sustained economical inefficiencies (David, 2007). To underline this argument, Arthur (1989) developed a simple analytical model using an extended stochastic Pólya urn, which will be briefly described subsequently.

2.1.1 Arthur's Extended Pólya Urn

In this model of path dependence, two types of agents (*R- and S-Agents*) are assumed to choose between two different technologies (A and B). The agents possess distinct natural preferences for each of the two technologies, and sequentially have to choose between adopting either technology A or B in random order. Equally distributed, half of them naturally prefer technology A (R-agents) and the other half prefer technology B (S-agents). A parameter *a* captures the preference of R-agents (a_R) and S-agents (a_s) for technology A, and a parameter *b* vice versa for technology B (b_R, b_s). Each agent bases its adoption choice on a payoff maximizing rule, which depends on its natural preference for one of the two technologies and on the number of agents who have previously adopted

[3] Reasons for increasing utility include network effects, learning effects or adaptive expectations. Section 2.1.2 describes these effects in detail.

technology A (n_A) or B (n_B). To what extent these previous adoptions influence an agents' payoff is captured by the two dissemination parameters r and s. For constant returns, the parameters are zero, for diminishing returns negative, and for increasing returns positive. Hence, in the case where the parameters are positive, the payoff for agents is increasing with the number of prior adoptions. As the sequence of agents adopting a technology is random, one type of agent may by random be overrepresented in the course of time, and hence its preferred technology may gain a lead. The advantage may be so striking, as to make the other type of agent switch towards this technology, because the payoff from prior adoptions exceeds the payoff received from their initial natural preference. Hence, once crossing a permissible boundary, both agents will choose the same technology, while the other technology will be locked out of the further technology adoption process. At the end of the adoption process, one technology will corner the market, because of the increasing returns and the contingent choices made by the adopters at the beginning of the adoption process. Figure 1 exemplarily depicts the adoption process under increasing returns, showing how the stochastic random walk property changes once a boundary is crossed. Here, after the lower boundary is crossed, both adopters choose technology B and exclude technology B from further adoption.

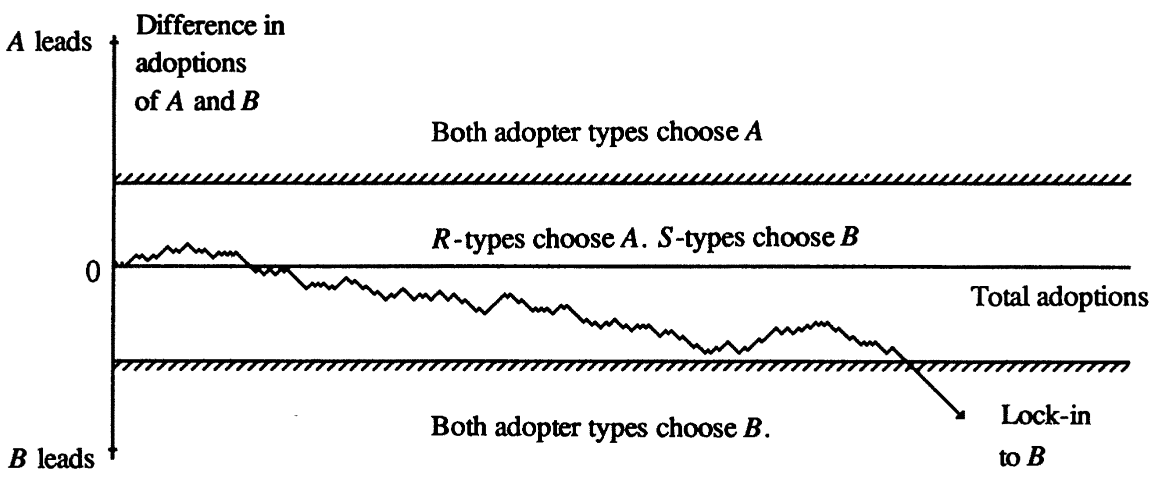


Figure 1: Sequential adoption process of two technologies by agents with different natural preferences (Arthur, 1989: 120)

Interestingly, as shown by Arthur (1989), this process outcome only holds in the case of increasing returns, while under a diminishing or constant returns regime the process will lead to an equally shared market between the two technologies. The properties of a path dependent process are hence governed by random events in the adoption sequence, as well as the increasing returns to adoption. Proving that this combination may indeed result in a potentially inefficient outcome, namely the chance that a superior technology is not adopted by utility maximizing agents, a simple stochastic Polyá urn model was extended by Arthur (1989) to depict a path dependent process.

The general purpose of conducting stochastic urn experiments is to make propositions about how chance and probabilities influence the outcome of a time-dependent process (Johnson, et al., 2005). A special case is the Pólya urn, which is particularly suitable to describe self-reinforcing processes (Mahmoud, 2009). Basically, the Pólya urn can be perceived as a container initially filled with one red and one black ball. From there, the procedure is as follows: randomly one ball is drawn from the container in each discrete time step, its color is observed, and then put back into the container with an additional ball of the same color. Drawing a black ball from the container at the beginning, for example, increases the probability of drawing black in the next round to two thirds.[4] By continuously repeating this drawing process, the proportion between red and black balls eventually approaches one out of an infinitely number of equilibrium states (Page, 2012). For greater clarity, the Pólya urn process is simulated for one hundred consecutive draws and six iterations according to the guideline of Hand (2006). Figure 2 depicts the results of this Pólya urn process. Even if not a representative sample, the results in Figure 2 show the importance of early draws. After approximately fifty draws, the process converges towards an equilibrium state, making the drawing probabilities more predictable. While the results confirm the importance of small events at the beginning, the Polyá urn does not reflect the case of a dominant technology and market lock-in (Arthur, 1983). Instead, the

[4] More generally, when at time t a total of x balls are drawn and of these x balls b are black then the probability of drawing a black ball at time t+1 is (b+1)/(t+2) (S. M. Ross, 2009).

shares converge in the Polyá urn towards a stable equilibrium. In order to reflect the lock-in phenomenon, the Pólya urn needs to be extended by the nonlinearities inherent in path dependence (Arthur, 1989).

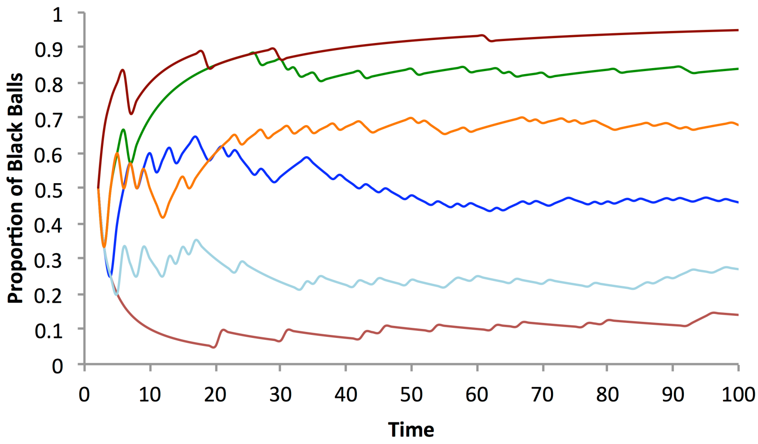


Figure 2: Own calculations for proportion of black balls in the Pólya urn using Excel

In contrast to the linear Polyá urn, drawing a ball in a non-linear urn increases the probability of drawing the same color in the next round disproportionately. Due to that, the process rapidly moves away from an initially unstable equilibrium towards one of only two stable equilibriums, eventually remaining there indefinitely. In these stable equilibriums, the probability of drawing a ball of a certain color is either zero or one, hence integrating lock-in. Figure 3 shows the drawing probabilities for a non-linear Polyá urn, computed with the Excel spreadsheet tool according to Hand (2006), for six iterations and one hundred consecutive draws. The urn model now depicts path dependence by including the draws of the balls as random small events, increasing returns in the drawing process through the non-linear behavior, and a lock-in into one out of two stable equilibriums as the outcome of the process.

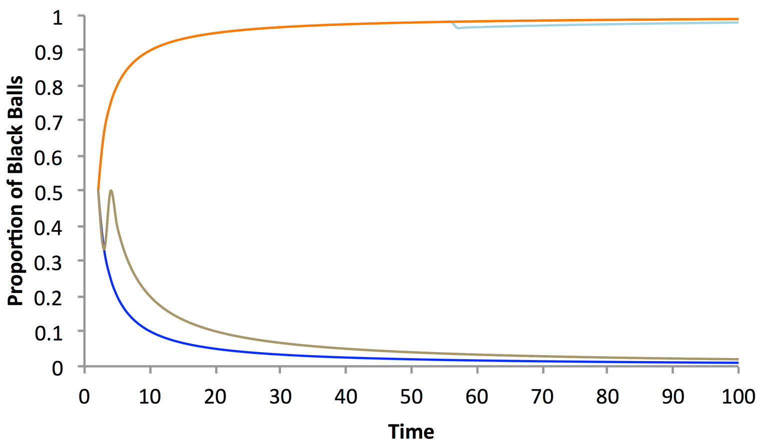


Figure 3: Own calculations for the extended non-linear Polyá urn using Excel

Based on this representation of path dependence in technology adoption, properties for a path dependent process have been derived, defining the presence or absence of a technological path (Arthur, 1989). Also, the case of increasing returns allows to distinguish the process from similar concepts, because it exhibits some special properties (Sydow, et al., 2009). The properties of an increasing return regime are listed in Table 1 and compared to the properties of constant and diminishing return regimes.

Table 1: Properties of constant, diminishing and increasing returns (Arthur, 1989: 121)

	Predictable	**Flexible**	**Ergodic**	**Necessarily path-efficient**
Constant returns	Yes	No	Yes	Yes
Diminishing returns	Yes	Yes	Yes	Yes
Increasing returns	No	No	No	No

As stated in Table 1, Arthur attributes *non-predictability, inflexibility, non-ergodicity,* and *possible inefficiency* to a path dependent process, which distinguishes the increasing returns case from other return regimes (Arthur, 1989).[5] As these properties are eminently important for the development and outcome of the process, and therefore the definition of path dependence, they will be briefly discussed.

Non-Predictability

Since random small events determine the outcome of the process, it cannot be predicted which technology is going to dominate the market based on the initial conditions. In fact, the definition of randomness already includes the lack of predictability. A truly contingent sequence does not exhibit a distinctive kind of pattern and therefore contains no information about future events (Beltrami, 1999). However, more precisely, a path dependent adoption process only exhibits the property of non-predictability in the first steps of the process as the urn stores information about future draws through the information pattern induced by the increasing returns. As a result, if the process converges towards one of the absorbing barriers, it becomes increasingly predictable (and in the case of lock-in further adoption is inevitable predictable). This is because the probability of drawing a locked-in colored ball equals one and because no diversity is present within the urn. By implication, a lack of diversity implicates predictability (Page, 2012).

Inflexibility

Once a technology is locked in, necessary policies or adjustments to unlock the technology increase without natural bounds. Due to the fact that increasing returns are not diminishing with the number of adopters in the extended Pólya urn model, the payoff for adopting a locked in technology approaches infinity under the assumption of an unlimited time horizon. Hence, to unlock or reverse the process,

[5] Arthur (1989) shows that under a regime of constant and diminishing returns the market is equally shared by the two technologies and lock-in does not occur.

the necessary adjustment has to match the sum of previous payoffs for the currently locked in technology. In other words, with regard to the extended Pólya urn the magnitude of adjustment needs to match or at least converge towards the number of balls from the dominant color to allow unlocking. It is obvious that this is hardly possible if the number of balls from the dominant color approaches infinity. Although a model is always an abstraction, it can clearly be doubted that in reality returns are increasing to infinity. Assuming that a critical mass is achieved, the utility of a technology more than likely approaches an upper boundary where increasing returns tear off and become constant or diminishing (Rogers, 1991). Still, even when increasing returns are bound, adjustments can be too expensive or hard to implement, so that the outcome cannot be altered when put into practice.

Non-Ergodicity

In mathematics, non-ergodicity is defined by Birkhoff's ergodic theorem stating that the average of a time function has to be identical to the average of the associated space function in order to call a process ergodic (Birkhoff, 1931; DasGupta, 2008).[6] In particular, the average of all equilibriums must be the same as the average of different initial conditions equating to a process not being affected by its initial conditions. Vice versa, non-ergodic processes produce different outcomes for different initial conditions. Instead of one single history, the initial conditions can alter the long-term outcome and generate multiple histories. The process itself becomes dependent on its history and develops like a path where historical events influence future outcomes.

Not Necessarily Path-Efficient

When increasing returns of adoption and contingency are at work, the process may favor and lock-in inferior outcomes. On the other hand, according to the definition of Arthur (1994), a path dependent process can also be efficient when

[6] See DasGupta (2008) for a mathematical definition and proof of Birkhoff's ergodic theorem.

the superior technology is chosen by chance right from the beginning and locked-in. Therefore, the concept of path dependence originally emphasizes that it is possible to lock into inefficient solutions; inefficiency, however, is not a necessary condition to call a process path dependent (David, 2007). Nevertheless, the chance that inefficiencies may potentially arise although individual choices are rational contrasts the neo-classical economics tradition (Arthur, 1994; Liebowitz & Margolis, 1990).

Carving out the properties of a path dependent process from a theoretical model is of great help to understand how paths develop, but proving that path dependence is indeed a real world phenomenon demands empirical evidence. A main critique is attributed to the concern that paths are purely theoretical and not observable in modern economies (Liebowitz & Margolis, 2013). Hence, to point out the empirical relevance of the theoretical definition of path dependence, several historical studies on technology diffusion building on the mentioned properties of path dependence have been conducted. Subsequently, some of the case studies examining path dependence are mentioned and discussed to convince the reader that paths emerge in economies and are indeed an observable phenomenon.

2.1.2 From the Pólya Urn to the Real World

The most prominent empirical example[7] of path dependence in the adoption of technologies is the history of the QWERTY keyboard design, narrated by Paul David (1985). David describes the technological diffusion process along a timeline, starting from the development of the typewriter up to the modern computer age. The story goes like this: In 1867, a printer from Wisconsin filed patent for a typewriter with a four-row keyboard design that prevented jamming. Because it displayed the letters QWERTY in the top-row, this design is known as the QWERTY keyboard. Although the early years proved to be hard, as new competitors with alternative keyboard designs entered the market, the QWERTY

[7] According to Google Scholar (http://scholar.google.de/scholar?q=QWERTY) Paul David's "Clio and the Economics of QWERTY" was cited 5398 times (as of 26/2/2013).

design rapidly evolved in the 1890s. Eventually, it became a worldwide standard for keyboard designs. According to David (1985), the main reasons for QWERTY achieving market monopoly could be attributed to economies of scale, complementarities between technologies, and irreversibility of past choices. A success factor for the QWERTY keyboard was, for example, the complementarities between typists and keyboard designs. As typists had to acquire machine writing skills through extensive training, a standard keyboard design decreases costs, as they do not have to learn typing on different keyboard designs and companies could then draw from a larger pool of trained typists. Although, today there is no technical reason for not switching towards another, possibly more efficient, design, most keyboards still assemble the letters QWERTY in the top row.

Building upon the historical narrative and case study methodology, researchers also extend the argument of path dependence to the evolution of other complex technologies (Arthur, 2009). Examples include the market dominance of the Video Home System (Cusumano, et al., 1992), the French (AZERTY) and German (QWERTZ) keyboard designs (Reinstaller & Hoelzl, 2009), chemical control of agricultural pests (Cowan & Gunby, 1996), railway gauges tracks in the American railroad industry (Puffert, 2002), quadraphonic sound systems (Postrel, 1990), gasoline engines for automobiles (Kirsch, 1996), alternating current electricity transmission technologies (David & Bunn, 1988), and light water nuclear reactors (Cowan, 1990).

Furthermore, the studies describe the self-reinforcing mechanisms responsible for the assertion of a dominant technology in detail. In the literature on path dependence, four of these self-reinforcing mechanisms are widely known: economies of scale, network externalities, learning effects, and adaptive expectations (Sydow, et al., 2009). In the following, the four mechanisms are briefly explained.

Economies of Scale

In economies of scale, an increase in the production output decreases the costs per unit made (Silvestre, 1987). Reasons for the decline in costs per unit can be the distribution of overhead costs on a larger number of output units, improved weighted average cost of capital, better exploitation of production capacities, or avoidance of indivisibilities in manufacturing equipment (Boyes & Melvin, 2008). Because of this, a product can be positioned on the market at a lower price, potentially increasing demand of consumers (D. D. Friedman, 1986). Increasing demand leads to higher production output, and therefore reinforces the cost down effect of economies of scale. This positive feedback loop may repeat itself, until the production costs per unit increase due to disposability of input goods or increasing coordination costs (diseconomies of scale). So, different to the increasing returns regime of Arthur (1989), the benefits of economies of scale are assumed to diminish over time. A good example is the supply of utilities, such as gas or electricity. For instance, real prices for electricity have dropped by 98% over a period of one hundred years between 1900 and 2000. The decrease in electricity production costs lead, over a self-reinforcing process, to the prevalence of electrical powered goods, such as lighting, telephones, or computers. Therefore, economies of scale also resulted in network effects. But, due to the external effects of electricity production, and the scarcity of raw materials used for energy production, prices are starting to increase again (Smil, 2006); economies of scale switched towards diseconomies of scale.

Network Effects

Network effects are described as a change in the utility an economic agent derives from a good, due to the fact that others are also consuming or using the same or a similar good (Farrell & Saloner, 1986; M. Katz & Shapiro, 1985; Leibenstein, 1971). A network effect can be considered positive when the benefit of a single user or consumer is greater the larger the number of previous adopters. This is the case in the Polyá urn model of Arthur (1989). Here, agents decide on basis of prior adoptions to maximize their utility. A further distinction can be drawn between direct, indirect, and two-sided network effects (Liebowitz & Margolis, 1994; Meyer, 2012; Parker & van Alstyne, 2005).[8] Direct network effects are present when an increase in usage of a service or good increases the benefit for an agent adopting this service or good. Numerous examples for direct network effects can be found in the communication and information industry. These include technological achievements, such as the telefax, landline telephones, mobile phones, or social networks, like Facebook or Linked-In. All of these services or goods have in common, that the benefit for a single user adopting one of these services or goods strongly depends on the installed user base. There would not be any utility or advantage in using a communication technology when there is no one to connect to. Indirect network effects are present when the number of available complementary goods increases with the size of the network. For example, it can reasonably be argued that the individual value of owning a home video player strongly depends on the number of blockbuster movies available for the video system. Further on, the number of blockbuster movies indirectly depends on the sold units of a video system. This is, because film studios will favor the device with most reach, and might distribute their products on this platform exclusively. Therefore, the individual benefit of an agent is only indirectly linked with the network size, as the distributors of movies determine to a great extent the attractiveness of a video system. In two-sided network effects the value of a network increases depending on the usage of two or more distinct groups. Take for example the diffusion of application enabled smartphones (Meyer, 2012). Consumers value the number of applications available for a specific operating system, and may base their smartphone purchasing decision on the size of an application ecosystem. At the same time, software developers value the number of previous adopters of said operating system. This is, because adopters of an operating system are potential customers for application developers. Comparable to a chicken-egg problem, consumers adopt smartphones with a specific operating system when a large amount of applications are available to download. On the

[8] Although literature suggests a variety of network effects two-sided, direct and indirect network effects are most commonly mentioned. See for example Sundararajan (2008).

other hand, developers only write applications if they are confident that enough users will download their app in order to make a profit.[9] Depending on the development of the user and app developer base, a specific platform can prevail and lock out competitors (Meyer, 2012). So, while size is important for all three types of network effects, interaction patterns to derive an individual utility might differ. For direct network effects, only adopters interact with each other, while for indirect and two-sided network effects, adopters interact with other groups.

Learning Effects

As an agent learns about a technology, by using it or through learning-by-doing, the technology might improve, and hence the benefit for adopting the said technology increases (K. J. Arrow, 1962). A self-reinforcing cycle, consisting of learning and further adoption, will eventually lock out other technologies, even if superior in the long run. Cowan & Gunby (1996) provide an example for learning effects by referring to the evolution of chemical pest control strategies in the agricultural industry. Prior to World War II biological, chemical, and cultural pest control strategies were equally applied. In the 1940s, the development of low-cost and effective chemical insecticides containing dichlorodiphenyltrichloroethane (DDT), which is now also known for its detrimental effects on the environment, altered the market situation. DDT rapidly replaced competing pest control techniques, not only in usage, but also in terms of research and development spending. Methods to improve chemical pesticides, instead of biological or cultural pest control strategies, were put into focus. Besides the effects of DDT on the environment, there is some evidence that chemical pesticides were the wrong choice with regard to the effectiveness of pest control. Instead, an integrated pest management, consisting of a mix between biological and cultural pest technologies, could be superior today, if similar learning effects were applied to the evolution of the technology. But, learning effects that occurred on different levels, make

[9] A recent discussion on the diffusion of smartphone systems in two-sided markets can be found in Meyer (2012).

switching towards alternative pest controls difficult. First, at the individual level, farmers learned how to apply chemical pesticides and made specific investments for chemical pesticides. Switching towards another technology would need a process of unlearning, and therefore new investments (Starbuck, 1996). Second, on a global, respectively regional, level, learning effects, due to research and development efforts in chemical pesticides, installed monitoring systems, and declining production costs resulted in learning curve effects. So, learning effects on different levels contributed to the lock-in of a doubtful pest control technology and made switching to other options difficult (Cowan & Gunby, 1996).

Adaptive Expectations

At last, a mechanism known as a driver for path dependence is adaptive expectations. In adaptive expectations, agents form expectations about the diffusion of a technology. By adapting their own expectations towards the expectations of other agents, uncertainties inherent in future events are supposed to be reduced. Therefore, the adaptation of a technology is not only influenced by current individual benefits of adoption, but also depends on the expectation of which technology will dominate the market in the future (Beyer, 2010). In particular, with regard to technological standards and technologies exhibiting "winner-takes-it-all" properties, adopters might align their decision for a technology towards the expected choices of other adopters (M. A. Schilling, 2002). Like a self-fulfilling prophecy, the faith in the superiority of a technology today, determines the prevalence of the technology in future. An example of adaptive expectations leading to the failure of a superior technology is Sony's digital audio tape (DAT) technology.[10] Although, experts emphasized the advantages of DAT, compared to conventional compact cassette systems, one reason for consumers not adopting DAT can be found in the formation of negative adaptive expectations. As consumers doubted that the technology might become an industry-wide standard for recording and playing audio, they did not switch from the compact cassette

[10] See Horner (1991) for an extensive case study of the digital audio tape.

technology to the digital technology. In a winner-takes-it all market, like audio recording, it is crucial to establish a standard in order to position a technology successfully on the market (Hill, 1997). Ignoring the superior audio quality of DAT, the negative expectations of consumers lead to the decline of the digital audio tape and instead, consumers quickly adopted the compact disc technology as successor of the compact cassette (Gandal, et al., 2000).

As shown, researchers point to these four different self-reinforcing mechanisms as drivers for path dependence, and convincingly argue, using empirical evidence, that lock-in of technologies occurs. Nevertheless, a dispute in the economic research community has risen, attacking the core of the path dependence concept. As path dependence itself is an attack on core assumptions of neoclassical economics, critics doubted that these empirical studies really show cases of strong path dependence. If the critics were to be believed, research dealing with path dependence would be doubtful or even obsolete. Therefore, this argument needs to be further discussed.

2.1.3 The Dispute about Path Dependence

To most researchers of path dependence in organizations or technologies, the QWERTY keyboard design is seriously flawed, when compared to the Dvorak keyboard design (Arthur, 1989; Castaldi & Dosi, 2005; David, 1985; Sydow, et al., 2009). In terms of typing speed and ergonomics, the Dvorak keyboard is, according to studies of path researchers, considered to be superior. But, the debate about the superiority of the Dvorak keyboard is still controversial, as its superiority, and the studies on which the claim of superiority is based, is vehemently questioned (Liebowitz & Margolis, 1994, 2013). Among these critics, Stan Liebowitz and Stephen Margolis seem to be the most persistent in emphasizing that the concept of path dependence is, as told by Arthur (1989) and David (1985), incorrect, or at least incomplete. Liebowitz & Margolis (2013) claim that small events and increasing returns on their own, as well as combined, cannot produce inefficient outcomes for an economy, at least not to a significant extent as described by David (1985). In the course of the past two decades, Liebowitz and Margolis have published several papers which question the inefficiency argument of path dependence, and react to Arthur's and David's harsh[11] remarks (David, 1997, 1999, 2001, 2007; Liebowitz & Margolis, 1990, 1994, 1995a, 1995b, 1996, 1999, 2013). To be precise, Liebowitz & Margolis do not question the existence of inefficiencies in markets, but instead highlight that increasing returns are just one constituting factor. Yet, increasing returns alone are not an accurate representation for mechanisms resulting in market failure and inefficiencies (Liebowitz & Margolis, 2013: 10):

"...a market to be locked in to something that is widely understood to be inferior requires more than just increasing returns or network effects, but in addition that an array of potentially profitable internalizing activities fail. These extra conditions imply that the kinds of failures that David and Arthur predict, though possible, are likely to be uncommon or of little economic importance."

In order to put more emphasize on this statement, Liebowitz & Margolis (1995b) developed a classified definition of path dependence, segmented according to their severeness of three different degrees of lock-in. *First-degree path dependence* describes a situation where the choices made by actors yield to an efficient outcome. Here, the best solution is chosen during the path process and locked in. This is comparable to the case of Arthur's model, when initial small events and self-reinforcing effects favor the superior technology. There is no problem for neoclassical economics present in this case, as the market had selected the most efficient solution by locking out inferior technologies. On the other hand, it is in hindsight almost impossible to prove that the locked in technology is superior to other technologies (Gould, 2002). *Second order path dependence* refers to a situation where an inferior solution is locked in, but actors have chosen the best possible solution present at the time of their decision. Even though economic losses occur, the actors could not have done better, because

[11] One of David's remarks on the inefficiency criticism of QWERTY (David, 1999: 9): "*But, to suppose that it is substantively crucial to any of the interesting issues surround path dependence and its economic policy implications is just plain silly".*

there was no superior solution available at this time, or agents did not know about this superior solution. *Third degree path dependence* exhibits inefficiency as actors could have known, or even did know, about the inferiority of the solution, but still have chosen to adopt it. In this case, lock-in could have been avoided by adopting agents. QWERTY is an example of third degree path dependence, according to David (1985). While the first and second-degree definitions of path dependence are consistent with regard to neoclassical theory, it is the third degree path dependence causing concerns for economists. Therefore, Liebowitz and Margolis deny the existence of third degree path dependence with noteworthy economic inefficiencies, and criticize the narratives on QWERTY and other technologies like VHS (Liebowitz & Margolis, 2013). Possibly, because of the substantial critique, Paul David eventually revised claims that path dependent processes need to be inefficient[12] or have to include increasing returns to adoption, but broadly defined path dependence as *"a dynamical process whose evolution is governed by its own history*" (David, 2007: 92) with *"no necessary connection exist*[ing] *between conventionally defined "increasing returns" and the phenomenon of path dependence"* (David, 2007: 102). While this statement again emphasizes that the inefficiency criteria is not a necessary construct in the definition of path dependence, defining the concept solely as a historical process would ignore the constituting properties mentioned by Arthur (1989). Therefore, the necessary propositions of contingency and self-reinforcement should be included in order to differentiate path dependence from similar concepts, and to prevent it from being stunted to a "*history matters"* argument (Sydow, et al., 2009). Thusly, even if inefficiency is absent, path dependence remains an interesting phenomenon, because it highlights how random events influence economic outcomes and may lead to market monopoly (David, 2007).

Despite the criticism, the concept of path dependence has been well established in literature on technology adoption (Vergne, 2013). The findings also had far

[12] David (1997: 8): *"Most prominent among the misapprehensions that have emerged in the literature during the past decade, at least to my way of thinking, is the notion that the condition of "path dependence" somehow is responsible for "market failures" which, in turn, result in persisting irremediable inefficiencies in the allocation of resources."*

reaching consequences in other domains apart from economics (March, 1991; Martin & Sunley, 2010; Pierson, 2000). Management scholars adopted the concept, for instance to describe resistance to change in organizations. Although management scholars refer to Paul David and Brian Arthur, attention has to be drawn regarding the peculiarities of social systems in order to transfer the concept of technological paths to an organizational context.

2.2 From Technologies to Organizations

Various literature streams, such as population ecology (Carroll & Harrison, 1994), evolutionary economics (Dosi, et al., 2011; Nelson & Winter, 1982), or the behavioral theory of the firm (March, 1991), pick up the notion of organizational path dependence, and argue that routines, standard operating procedures, social interactions, communication patterns, or (behavioral) rules in organizations evolve path dependent and eventually impede change processes. But, while the concept is prominently adopted, theoretical grounding of path dependence in organizations still remained vague in these literature streams. In a recent endeavor to overcome these theoretical difficulties, a framework for organizational path dependence has been developed by Sydow et al. (2009). Sydow et al. (2009) recognized that the concept of path dependence is inadequately defined in management literature, and broadly used as an argument for how *"history matters"* in organizations. As a remedy for this deplorable state, they suggested a process oriented theoretical framework, divided into three consecutive phases, to define how an organizational path constitutes itself, and to provide a guideline for examining path formation processes. Although, the framework is strongly related to the early definition of path dependence in technology adoption, there are some differences worth mentioning.

2.2.1 A Framework for Organizational Path Dependence

In the framework, the path formation process is divided into three consecutive development phases, where the set of path dependent properties defines and, at the same time, separates the phases from one another. Furthermore, instead of technologies, the objects of observation are strategic options, or the range of variety in strategic decisions, a company may pursue in order to adapt to its environment.

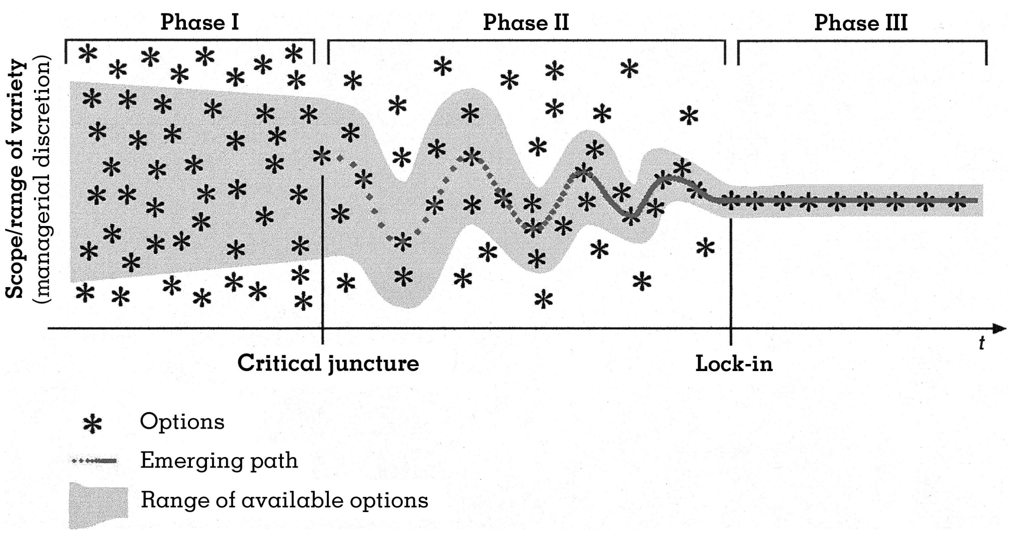


Figure 4: Three phases of organizational path dependence (Sydow, et al., 2009: 692)

In *Phase 1* an organization can choose among multiple options, while long-term consequences of choices are still unpredictable, because of uncertainties in the environment. However, the selection of one option may trigger a self-reinforcing process. In contrast to the initial situation of technology adoption, history matters in organizations (even in the early phase), while not determining the process right from the beginning. This is caused by the fact that organizations are rule guided social systems, composed of individuals with separate prior histories, and hence constraining the range of options (Kimberly, 1975). The grey shadow in Figure 4 visualizes this restriction, by excluding options that cannot be pursued by the organization. A simple example of eliminating strategic options is stating the business purpose, which is necessary to register a business in Germany. Even under the best conditions, it is hardly possible for an organization registered as a bakery to pursue business in the automotive sector. With transitioning to *Phase 2,* a self-reinforcing process shapes a path by progressively narrowing down the scope of strategic options an organization may pursue. Originating from a critical juncture, which can be conceived as a set of choices already made in the first phase, a dominant regime develops, favoring the reproduction of a same small set of choices. Potential drivers for self-reinforcing mechanisms are, for example, organizational learning effects (Levinthal, 1997). Still, switching to other options is possible, but may be difficult, because the process becomes exceedingly irreversible over time. Building on self-reinforcement, instead of increasing returns, the framework includes behavior not necessarily driven by utility maximization, such as uncertainty avoidance, power preservation, or reduction of cognitive dissonance (Sydow, et al., 2009). Eventually, the self-reinforcing mechanisms lead to *Phase 3,* where the set of choices become locked in and render the process inflexible. Switching to other options is now hardly possible for the organization, and as a result potential inefficiency is implied. Compared to technological paths, the lock-in is described as a predominate behavior, or as Sydow et al. (2009: 695) point out, "*an underlying core pattern"*. Different to the original definition is that there is at least some variation in organizational behavior, which in turn includes the notion of heterogeneity in organizations. Therefore, in contrast to the path dependence concept brought forward by Arthur (1989), the variation inherent in this definition of path dependence allows an organization to change, even though being locked-in. In the language of the Polyà urn, lock-in is now not a state where only one type of balls with one color is drawn from the urn, but there is a chance that another color might be drawn as well. Therefore, within this framework, the concept of path dependence is not only transferred to organizations, but also major adjustments have been made to the original concept. Besides introducing the possibility of variation, increasing returns are replaced with self-reinforcing mechanisms. Therefore, the self-reinforcing effects as drivers for an organizational path dependent process have to be adapted in order to account for the peculiarities of social processes. According to the literature, four different organizational mechanisms can broadly be distinguished (see Table 2).

Table 2: Self-reinforcing mechanisms in organizations according to Sydow, et al. (2009)

Mechanism	Remark
Coordination Effects	Adoption of the same rules, behaviors, or beliefs facilitates the interaction between individuals, as conformation reduces interaction costs and maintains internal consistency (D. Miller & Friesen, 1984). The self-reinforcement of conforming behavior results in fixed patterns of tasks, routines, or rules in an organization, which are hard to abandon in the presence of environmental change. Examples include the rule-guided behavior in the newspaper industry to produce high-quality content, instead of exploiting online markets (C. G. Gilbert, 2005; Koch, 2008), or the inability of Polaroid to change from a razor-blade towards a digital business model (Gavetti, 2005).
Complementary Effects	Combining interrelated rules creates synergies by either lowering the costs, or increasing the benefit, of practicing them jointly (Stieglitz & Heine, 2007). Therefore, complementary rules, routines, or practices may be self-reinforced and become dominant in an organization (Leonard-Barton, 1992). One example for complementary effects potentially inducing path dependence, is the interrelatedness between R&D and marketing capabilities of organizations, adding up to a core competence (Prahalad & Hamel, 1990; Sydow, et al., 2009).
Learning Effects	According to the concept of learning curves, the efficiency of performing a task increases with the number of subsequent iterations, as gained experience facilitates to perform the task more reliable and faster. Furthermore, decreasing costs, because of learning, increases the attractiveness of the solution and may hence impede switching to other learning domains. The refinement of current capabilities may then crowd out the search for new domains (March, 1991, 2006). Additionally, learning effects are often further reinforced by coordination and complementary effects (Sydow, et al., 2009).

Adaptive Expectation Effects	Because individuals in an organization are socially embedded, the formation of beliefs and preferences is influenced by the expectations of others around them. To fulfill the need to belong and for uncertainty reduction, individuals adapt the expected dominant belief set in order to be on the "*winning side*" (Sydow, et al., 2009).

In addition to including variation and replacing increasing returns with self-reinforcing mechanisms, two more differences worth mentioning are brought up by Sydow et al. (2009). First, the contingency assumption is softened compared to the extended Polyá urn and second, the notion of potential inefficiency is introduced. As contingency lies at the core of path dependence, changing this assumption might be problematic and therefore need further explanation.

Contingency in Organizational Path Dependence

To account for firm strategies, individual agency, and intentional behavior, Sydow et al. (2009: 693) propose, "*a less randomized modeling*" of path formation. Instead of describing individual behavior as an arrangement of random social events, agents are assumed to act on purpose and influence history through entrepreneurial actions (Garud & Karnoe, 2001). Ajzen's (1991) theory of planned behavior even goes as far as predicting intentional behavior from attitudes, norms, and perceived behavioral control. From this, it can be assumed that individual behavior is not random, but more or less predictable. Yet, it is not the behavior that is random, but the interactions of deliberate actions bringing about complex, unintended, and unexpected consequences, which may be understood to be unpredictable or perceived as nearly random events (Weick & Roberts, 1993). That is, even if agents act intentionally, the outcomes may not be predicted in advance, because of the complexity in social interactions. Prominent examples, where such deliberate actions result in contingent events, can be found in the path creation literature, especially in the cases of the development of Viagra (de Rond & Thietart, 2007) or Post-It Notes (Garud, et al., 2010).

Furthermore, it must be abundantly clear what is meant by the term "*less random*". If randomness is defined as the lack of predicting future events, because of the absence of an underlying information pattern,[13] a "less random" process would suggest the presence of at least some pattern, like in a pseudo-random sequence (Beltrami, 1999). This would infringe the original assumption that the beginning of a path dependent process is fully unpredictable, because an underlying pattern would at least allow for "some" prediction. To some extent, this assumption proves to be right, as history proceeds along pathways at least allowing for some prediction, even right from the historical origin (Hannan & Freedman, 1977). Therefore, maintaining the theoretical core of path dependence, I suggest including contingency in the following form: Agents in organizations make decisions deliberately, based on their prior experience. However, due to complex interactions, imperfect information, and bounded rationality, the consequences of these actions include some seemingly random elements that make accurate predictions near impossible. Even with the same initial conditions, these pseudo-random elements may push the process in different directions, ensuring that the process is not deterministic, but at the same time allow for at least some prediction.

Potential Inefficiency of Organizational Path Dependence

While Arthur (1989) and David (2007) state that a path dependent process does not necessarily need to be efficient[14], Sydow et al. (2009) argue that at least *potential inefficiency* must be present, because of an organizations inability to change. This potential inefficiency is severe enough to raise concerns about the lock-in state. Unlike inefficiency, *potential inefficiency* does not necessarily imply immediate economic loss, but is the mere inability to change when more efficient solutions are present, or might be present in the future (Holtmann, 2008). More

[13] Beltrami gives an overview of what randomness exactly means by, for example, referring to probability theory, information theory, determinism, and the perception of randomness. Even though a complete definition of randomness should therefore include more than just the absence of patterns and the inability to predict future states using today's observations, a precise definition would go beyond the scope of the dissertation at hand.
[14] But also do not exclude the case where increasing returns lead to an efficient lock-in.

accurately, inefficiency can be described by an organization's inability to change and is the outcome of a path dependent process (Petermann, et al., 2012). But, potential inefficiency is inherent in every organization and only depends on the severity of environmental change (Hannan & Freeman, 1984). Even highly efficient organizations may turn inefficient when facing tremendous environmental change, and may be unable to adapt. Furthermore, for organizations operating in a stable industry, the ability to adapt to all kinds of environmental changes might prove inefficient itself. Think of a match manufacturer A. Assume that matches are a homogenous good and customers are therefore very price sensitive. If the match manufacturer wanted to hedge against a change in consumer preferences, say that consumers prefer lighters instead of matches, it could invest in capabilities to produce lighters. Such investments may inflate the cost structure and that could lead to an increase in the prices of matches. If a competitor B does not hedge against the risk of changes in consumer preferences, but is locked-in to produce matches, it could potentially offer the matches cheaper compared to A. As matches are a homogenous good, customers would switch from A to B. In this scenario, flexibility proves to be inefficient compared to path dependence. Here, the lock-in of B with its *potential inefficiency* proves to be better than the hedging strategy of A, when the environment is stable. Because of that, the inefficiency criterion is somewhat ambiguous and depends on competition, environmental stability, and a point of reference. But, as David (2007) shows, path dependence remains interesting, even when not taking into account any kind of efficiency criterion. Because of that, organizational path dependence is defined left ajar on the definition of David (2007), and in accordance with the definition of Vergne & Durand (2010: 741):

"*as a property of a stochastic process which obtains under two conditions (contingency and self-reinforcement) and causes lock-in in the absence of* [an] *exogenous shock*".

This definition goes along with Sydow et al.'s model of path dependence, includes contingency, and excludes any efficiency condition. With this clear definition at hand, it is also possible to distinguish path dependence from related concepts in organization science such as imprinting (Stinchcombe, 1965), structural inertia (Hannan & Freeman, 1984), commitment (Ghemawat, 1991), institutionalizing (Powell & DiMaggio, 1991), reactive sequences (Mahoney, 2000), or escalating commitment (J. Ross & Straw, 1993).

Up to now, this view does only describe how organizations are trapped in a lock-in, but does not make any statements about how organizations can escape lock-ins. In the path dependence definition of Arthur (1989), David (1985), and the three-phase framework of Sydow et al. (2009), the lock-in phase is infinitely reenacted, necessarily causing the downfall of an organization in the presence of environmental change. Such an overly deterministic and mechanistic view of organizations negates free will and individual agency, which are definitely characteristics of individual behavior (Bourgeois III, 1984). Garud and Karnoe (2003) describe the creation of new paths even as a process of “mindful deviation”, where individuals possess agency and are acting according to their beliefs. A more complete framework should include agency, and therefore the possibility that paths are not perpetuated forever, but may be unlocked either accidentally or intentionally. In order to include how organizations can escape paths, the next section reviews prior literature on unlocking of paths in technologies and organizations.

2.3 The Unlocking of Path Dependence

Early research on path dependence focused on the assertion of inefficient technologies under a regime of increasing returns, while the notion of switching from one locked in technology towards a new technological path, or unlocking a technological path, only received little attention in the literature. One problem is that Arthur's (1989) extended Polyà urn, as the prevalent illustration of technological path processes, is unable to capture this dynamics. Here, when the number of balls with a designated color exceeds an upper or lower boundary, the probability of drawing the opposite color tends towards zero, because of the unboundedness from increasing returns (Arthur, 1989). Based on this representation of path dependence, locked in technologies remain unchanged in a stable equilibrium forever, thus negating the possibility of unlocking (Loch & Huberman, 1999). In conclusion, autonomy of actors is ignored, as the fate of a locked in technology cannot be altered in hindsight. But, as common sense suggests that "*no path is forever*" (Sydow, et al., 2009: 701), and convincing empirical cases are known where new technologies challenge market conditions, and sometimes even succeed, it can reasonably be assumed that the unlocking of technologies occurs (Martin & Sunley, 2010; Witt, 1997). Supporting this statement, Table 3 names a few selected empirical examples, directly linked to the literature on technological paths. It states empirical cases, where locked in technologies have been replaced, it is convincingly argued that technologies will be replaced in the near future, or that policies for escaping lock-ins were established. With regard to related literature in the field of innovation management, similar cases of radical and disruptive innovations breaking technological paths can be found, as for example solid state drives (Christensen, 1997), photolithographic alignment equipment (Henderson, 1993), or light emitting diodes (Sood & Tellis, 2011).

Table 3: Examples for unlocking of technological paths

Lock-in	Successor	Source
VHS	DVD	Dolfsma and Leydesdorff (2009)
Gasoline vehicles	Electric vehicles	Cowan and Hulten (1996)
35mm film	Memory cards	Munir and Matthew (2004)
CFC-Refrigerators	HFC-Refrigerators	Araujo and Harrison (2002)
Cassette Players	CD Players	Liebowitz and Margolis (1995c)
Fossil power plants	Renewable Energy	del Rio and Unruh (2007)

In conclusion, comparing the behavior of the Pólya urn with phenomena observed in the real world shows that they do not correspond when it comes to unlocking. Witt (1997: 762) puts the argument it in a nutshell by stating that:

"...there would be no point in speculating about detrimental effects of technological "lock-in", if there were no possibility of the situation being changed as the probability model of the generalized Pólya urn scheme literally claims. Experience teaches, of course, that, sooner or later, there will always be new rivals who threaten the market dominance of a technology or a variant."

Because of the overwhelming empirical evidence on unlocking, the assumption of complete inflexibility, as claimed by the Polyà urn model, must be denied. The *"overly static view of the social world"* (Pierson, 2000: 265) implied by the model is justifiably reproached by critics to be a strong weakness of the concept (Pierson, 2000; Rogers, 1991). In order to dispute the critics' point of view, this weakness has to be addressed. Therefore, the case of unlocking has to be included in the model, and necessary conditions have to be derived without giving up the core of path dependence; namely, self-reinforcement, contingency, and lock-in (Kuhn, 1962). David (2005: 187) gives a starting point for the inclusion of unlocking into a dynamic model of path dependence by hinting to the evolution of path dependence as a punctuated equilibrium process[15]:

"Sudden shifts in structure, corresponding to the new evolutionary biologists' notion of 'punctuated equilibria'... may open up a way for the formulation of dynamic models that are compatible with 'stage theories' of development"

Opposed to Darwin's evolutionary model, in which species gradually adapt to a changing environment, the punctuated equilibrium theory states that a long period of stability is interrupted by a sudden shock, where pressure for change prevails and a new species can evolve (Eldredge & Gould, 1972; Gould, 2007). Evolutionary biologists explain these discontinuities, resulting from a punctuated equilibrium process, by the existence of spatially isolated subgroups, evolving alongside a larger mother species. While the gene flow in the central mother

[15] See also Martin & Sunley (2010: 70).

population causes homogenization of the gene pool, these forces are sometimes too weak for preventing the emergence of heterogeneous local differentiations. Within these small subgroups, a decoupled selection mechanism may be effective, possibly developing the split local species in a different direction compared to the mother species. In the case of abrupt environmental change and integration of the small subgroup into the larger population, the genes of the small subgroup may be selected and outcompete the gene set of the mother species. Eventually, the genes of the once isolated group might prevail, and the mother species becomes extinct (Gould, 1980). Although, proof of evolution in accordance with a punctuated equilibrium process remains inconclusive and controversial, the concept was successfully applied to describe phenomena related to the diffusion of new technologies and path dependence (Dawkins, 1996; Loch & Huberman, 1999). In fact, the notion of punctuated equilibrium shares similarities with the concept of path dependence (A. L. Schneider, 2006; Schwartz, 2004). Both concepts assume that contingent historical events have a serious impact on the outcome, that the process is not determined from the beginning, that the scope of choice is restricted by prior history, and that the occurrence of a stable equilibrium causes rigidity and (as a consequence) resistance to change (Gersick, 1991). But, while the punctuated equilibrium theory explains how phases of stasis are dissolved and populations adapt to new environments, the original concept of path dependence does not take environmental changes into account. However, if we take David's (2007) advice, and understand path dependence as a punctuated equilibrium process, the case of unlocking may be convincingly integrated in the concept.

2.3.1 Unlocking Technological Paths

The first attempt to explain how unlocking of path dependence as a punctuated equilibrium process can occur was undertaken by the geographic economist Ron Martin, who distinguished three different types of evolutionary path models, that raised the notion of unlocking. Martin and Sunley (2010) name these types the Setterfield-Type, the Non-Equilibrium-Type, and the David-Type. The Setterfield-Type model assumes a broader definition of path dependence compared to the previous definitions. In the Setterfield-Type model, unlocking of paths may occur without the presence of an exogenous shock or any other type of environmental change (Setterfield, 1999). Instead, the lock-in is conceived as being only temporarily, and the system may escape lock-in by an *"endogenous process of innovating-out of equilibrium"* (Martin & Sunley, 2010: 17). So, instead of triggering an organizational transformation process, the organization itself may be responsible for environmental change, for example, by enacting their views on the environment. In a computer simulation model K. D. Miller and Lin (2010) show the arising dynamics when organizations enact their beliefs on the environment. By this, the role of economic actors, like entrepreneurs, purposefully looking for opportunities to change the current status is highlighted, being similar to the notion of path creation (Garud, et al., 2010). The second type, the Non-Equilibrium-Type, excludes the whole notion of lock-in from the definition of path dependence. Here, a path dependent process is described as consisting of alternating phases of rapid and gradual evolution without convergence towards any kind of equilibrium state. Instead, technologies or social systems are supposed to evolve along trajectories. Furthermore, these trajectories are not only shaped by their own history, but also through co-evolutionary processes by interacting with adjacent technologies, industries, or institutions. Based on this definition, path dependence degenerates to a *"dynamic open historical process"* (Martin & Sunley, 2010: 18), which in the end may be broken down to a simple "*history matters*" argument (Sydow, et al., 2009). While both of the prior models to some extent include the process of unlocking, the applied definition of path dependence is very broad and imprecise (Sydow, et al., 2005). A more restrictive model based on the original assumptions is thus advisable. The proposed David-Type model resembles the present definition of path dependence more closely, as a contingent process with increasing returns eventually locking into one of multiple possible stable equilibrium states. Extending the original definition of David by the notion of unlocking is achieved through including an exogenous shock in the environment as punctuation. This punctuation eventually de-stabilizes the system, and gives way to the selection of environments leading to the advent of new technologies or industries and emerging path dependent processes (Figure 5). The process of path formation, stable equilibrium, and punctuated unlocking is continuously repeated, making the underlying logic of the model akin to the notion of S-curves in technology adoption and product life cycles (Loch & Huberman, 1999; Rogers, 1962).

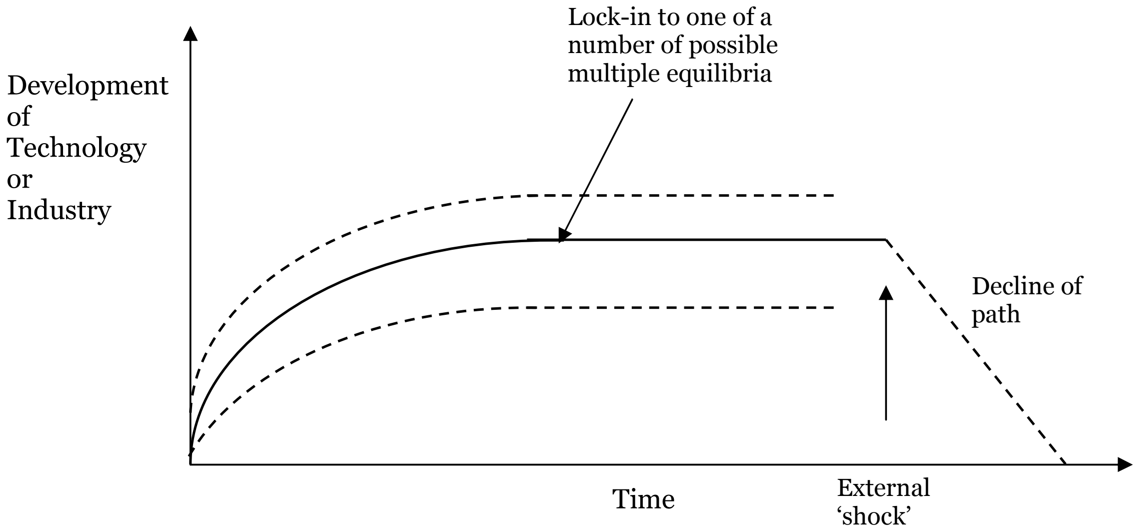


Figure 5: Unlocking as a punctuated equilibrium process (Martin & Sunley, 2010: 17)

An empirical example for a punctuated path dependent process with unlocking can be found in disk drive technologies (Christensen & Rosenbloom, 1995). Invented in the 1950s, the hard disk drive technology was until recently unrivalled for stationary data storage applications. With the advent of portable electronic technologies, such as digital cameras, solid-state storage technologies, with smaller form factor and shorter average access time, were developed to account for the peculiarities of mobile applications. Along the increasing need of mobile solutions for more memory, the storage size of solid state cards and drives increased. At the same time, prices per megabyte dropped to a point where it became feasible to use the storage technology for desktop or laptop computers. While nowadays hard disk drives are still mainly installed into desktop computers, solid-state disk drives abandoned their niche existence and, due to their advantages, may replace the computer hard disks in the long run. The lock in into the hard disk drive technology, and subsequent unlocking through the solid-state disk drives, is therefore corresponding to the punctuated equilibrium model. In the empirical example, the exogenous shock or punctuation can be attributed to the rise of digital cameras and MP3-players.

Similar to the formation of technological paths, it can be assumed that social systems like organizations exhibit some peculiarities with regard to the unlocking of paths. With the knowledge of unlocking technologies, through punctuated shocks in the process, the notion of unlocking organizational paths is in the following examined along the same line.

2.3.2 Unlocking Organizational Paths

Processes of punctuated equilibrium are not only observed in the diffusion of technologies, but also in social processes on individual, group, and organizational level (Gersick, 1991). Here, the punctuated equilibrium process provides an explanation for abrupt organizational changes after long phases of stasis (D. Miller & Friesen, 1984; Romanelli & Tushman, 1994; Tushman & Romanelli, 1985). In longitudinal case studies on organizations like ICI, AT&T, Citibank, General Radio, or Prime Computers, it is convincingly argued that firms reinforce their internal structure and strategy towards a stable equilibrium state (Pettigrew, 1987; Romanelli & Tushman, 1994). The phases of stable equilibrium are punctuated by short periods of rapid change, triggered through an exogenous shock in the environment. This results in significant disruptions of firm structure, strategy, organizational membership, and/or business processes, eventually leading to convergence towards a new stable equilibrium (Tushman, et al., 1986). According to this view on social change, organizations repeatedly traverse periods of stable equilibrium, incremental adaptation, and inertia, punctuated by profound change and strategic reorganizations (Tushman & Romanelli, 1985). For instance, IBM has continuously reinvented its strategy and product lines, evolving from a punch card equipment manufacturer over mainframe producer towards an integrated solution oriented infrastructure provider in the software and hardware industry (Hendron, et al., 2005; Maney, 2003). In punctuated equilibrium theory, organizational change is exogenously stimulated by environmental disruptions, potentially resulting in a decline of firm performance. Tushman & Romanelli (1985) further restrict this argument by stressing that only major or sustained declines in firm performance will trigger organizational transformation processes. Thus, it can be argued that radical change in the environment of an organization may evoke a fundamental change process.

Yet, the research stream on punctuated equilibrium in organizational change is only loosely linked to the concept of path dependence. Nevertheless, it exhibits significant similarities as path researchers also highlight the necessity of exogenous shocks or external interventions to escape lock-ins (Sydow, et al., 2005, 2009; Vergne & Durand, 2010). These exogenous shocks may open up a *"window of opportunity"* to escape harmful lock-in situations, for example by initiating entrepreneurial actions (Burgelman & Grove, 1996; Castaldi & Dosi, 2005). A first step in integrating the unlocking of paths into an organizational definition of path dependence was taken by Vergne & Durand (2011). Vergne & Durand (2011: 372) are not only precise in defining lock-in, but also in defining the necessary conditions for escaping paths in organizations:

"Management scholars can think of lock-in as an organizational situation that can be altered only at a prohibitive cost and in response to strong exogenous pressures (e.g. economic crisis, radical technological change, political turmoil)."

Just as in punctuated equilibrium theory, it is assumed that a radical exogenous shock is needed for unlocking organizational paths, as it puts strong pressure on the organization.[16] This pressure might come from stakeholders outside of the

[16] Deviating from this view Pentland et al. (2011) show how path-dependent routines may cause endogenous change through variation. Within the model they define path dependence as a "*process through which past actions influence the likelihood of future actions"* (Pentland, et al., 2011: 1490). As in the Setterfield-Type and Non-Equilibrium-Type model, this does not necessarily include the notion of lock-in.

organization or from people within the organization. For example, individuals within or outside of the organization may challenge the status quo of organizational leaders because of declining firm performance (Greve, 1998; Hendron, et al., 2005). As a result, new dominant coalitions may form when established power structures corrode (Cyert & March, 1963). Like in the punctuated equilibrium theory, this statement agrees that an exogenous shock needs to trigger an organizational response for initiating necessary change processes. Apparently Vergne & Durand (2011) do not have an opinion on how this response must look like. According to Sydow et al. (2009), one could think of two possibilities for escaping organizational paths. Either the path is intentionally broken, for example by a change or transformation program, or actors in the organization stop reproducing the underlying pattern, which leads to the dissolution of so called *"deep structures"* and organizational paths (Gersick, 1991). The intention of actions to escape paths is therefore relevant to distinguish between path dissolution and path breaking. To explain the difference, and as a mean of illustration, one empirical example for path dissolution and one for path breaking is given subsequently.

Path Dissolution

A well-known example for path dissolution can be found in the case of Intel's transformation from a computer memory company into a microprocessor company (Burgelman, 1994; Burgelman & Grove, 1996; Sydow, et al., 2005). Remarkably, the organizational transformation process was not initiated by the top management team, but members of the middle management helped to overcome Intel's strategic disorientation (Burgelman & Grove, 1996; Inkpen & Choudhury, 1995). Following a simple decision rule, the middle management used the flexibility in their scope of action and ramped up production capacities for microprocessors at the cost of memory chip production. While the formal internal structure of the organization remained intact, change was nevertheless initiated through the adaptation and reconfiguration of established routines to the new environment. Eventually, the top management became aware of the importance of microprocessors for the future of Intel, adjusted the strategy, and aligned the organization towards the new direction by restoring external fit (Siggelkow, 2001).

An important factor for the dissolution of the memory path at Intel was the internal selection environment, allowing for the reallocation of resources, and the organizational culture, tolerating dissonance within the organization. Nevertheless, it is the task of the top management to value dissent in the organization, and to find a balance between the bottom-up and top-down forces in strategic management (Burgelman & Grove, 1996). As shown in the Intel case, it can be the same rules, routines, or mechanisms that lead to the organizational path in the first place that help an organization to escape lock-ins and make it an adaptive learning organization (Dodgson, 1993). While the dissolution of paths is an emergent process, breaking path dependencies includes the realization on the top management level for the need to change.

Breaking Paths

An example of breaking path dependencies in organizations can be found in Liz Claiborne, a manufacturer and retailer in the apparel industry (Siggelkow, 2001). After a very successful period of growth and high profits in the 1980s, the company faced a serious decline in net income at the beginning of the 1990s. Reasons for the financial downturn were crucial changes in consumer preferences, product portfolio, and distribution channels. Although the shift in the environment was detrimental to the firm performance, the management team did not initiate necessary radical changes. Instead the top management preserved the internal fit by only changing the organization incrementally towards the new environment. Even worse, the strong internal culture and past success reinforced the confidence in the status quo. With the situation further exacerbating, an outsider COO was hired in 1994, eventually succeeding the CEO of Liz Claiborne in 1995. After the succession, the new CEO replaced the majority of the top management team with newly hired managers from outside the organization. Top management turnover is a common response for organizations facing environmental change, and is often used to import new mental maps, to allow for second-order learning processes, and to unlearn gridlocked organizational routines (Lant, et al., 1992; Nystrom & Starbuck, 1984; Siggelkow, 2001). Indeed, the new top management team realized the need for reconfiguration, and performed wide-ranging changes in design, product portfolio, distribution, manufacturing, and product presentation. These changes were fit destroying, leading to a broadening of the scope of action and revitalization of Liz Claiborne that finally unlocked the organizational path. So in contrast to path dissolution the intentional actions of the top management to reconfigure the organization and change the strategy were decisive.

While a clear distinction between the dissolution and breaking of organizational paths can be made, most of the literature on path dependence does not completely or does only insufficiently differentiate between these two constructs. Instead the literature on path dependence refers to the notion of de-locking, unlocking or breaking paths (Castaldi & Dosi, 2005; Ericson & Lundin, 2013; Hassink, 2005). In a similar vein this dissertation uses the unifying term of unlocking instead of differentiating between path dissolution and path breaking. The reason for this approach is that the dissertation focuses on how unlocking can occur on an organizational level and does not want to explain the intention of actions that lead to the unlocking of paths. Nevertheless it is important to know, that means of unlocking can be classified into path dissolution and path breaking. Subsequently the literature on means for unlocking paths is revisited to get an understanding of how organizations might potentially escape paths.

2.3.3 Means for Unlocking Organizational Paths

Unlocking is here simply being defined as an interruption of the dominant self-reinforcing logic that brought about path dependence in the first place. Furthermore, this interruption has to open the scope of action for implementing a superior alternative (Sydow, et al., 2009). An alternative can be considered superior if the implementation of the alternative leads to higher firm performance; or in other words, a better adaptation towards a changed environment compared to the status quo. Despite the potentially self-destructive effects of path dependence on organizations, the current research on how an organization or individuals in an organization should respond in order to unlock organizational paths, at least in a narrow sense, and adapt to the changes in the environment is according to Ericson and Lundin (2013) in a very early stage. Still, evidence on how these means influence organizational paths is missing. Table 4 provides an overview of means to unlock organizational paths proposed so far in the literature.

Table 4: Review of literature on mechanisms to unlock organizational paths

Mean	Remark	Source
Invasion	Invasions happen when cultural traits, organizational forms or individual beliefs that have been developed somewhere else diffuse into a social system. Such invasions occurred for example in modern American history. Here, conquistadors colonized America through invading territories of indigenous peoples. With regard to organizations invasions are usually less violent and refer to the emergence and adoption of, for example, management fashions, new ways of organizing, mergers & acquisitions, top management team turnover or cultural changes. Invasions force an organization to adapt to the new circumstances. Examples for invasions in organizations can be found in the introduction of tayloristic working principles, corporate social responsibility programs, lean manufacturing or total quality management (Castaldi & Dosi, 2005). Pressure for adopting new management principles may come from stakeholders, brought to the company by business consultants or implemented by the management itself (Sarkis, et al., 2010). In particular external consultants can be a powerful mean to uncover deadlocked routines, irritate the system through interventions and put path dependence on the strategic agenda (Sydow, et al., 2005).	Castaldi and Dosi (2005); Karim and Mitchell (2000); Mahoney (2001); Spell (1999); Sydow, et al. (2005); R. Williams (2004)

Cognitive Dissonance	A gap between the prescribed role of an individual, established social norms, or expectations and individual mental models, identities, or self-perceptions may accrue. This gap results in a conflict, named cognitive dissonance. Assuming that individuals have control over their beliefs they could either reduce cognitive dissonance by adapting their beliefs, manipulate others to change their beliefs or assign low importance to the dissonant beliefs. The second-case may initiate a second-order or double-loop learning process in the organization and can be a strong driver for unlocking paths (Argyris & Schoen, 1978).	Akerlof and Dickens (1983); Burgelman and Grove (1996); Castaldi and Dosi (2005)
Diverging behavior	Deviant behavior of individuals in organizations may prevail against the self-sustaining forces of path-dependence. Eventually deviant behavior may cause a chain reaction in the organization and lead to new organizational structures. Castaldi & Dosi (2005) compare this behavior to the physical phenomenon of "symmetry breaking" where small deflections are decisive for the state of the whole system. An example for deviant behavior can be found in the corporate entrepreneurship literature with the "Post-it" case. The invention goes back to Spencer Silver, who accidentally invented a weak adhesive instead of super glue within a lab of 3M. Although diverging from the dominant logic of the company that adhesives have to be strong, he was able to mobilize resources and promote the glue successfully within the company. Art Fry eventually had the idea to use the weak glue to hold bookmarks. These bookmarks are now known as "Post-it" notes.	Allen (1988); Castaldi and Dosi (2005); Garud, et al. (2010)
By-product of path formation	In this mean to unlock paths, the same mechanisms leading to lock-in in the first place are also responsible for the unlocking of organizational paths. Organizational routines, behavioral patterns, interaction structures or decision rules may be selected over multiple co-evolving selection domains. Changes in one of these selection domains may entail a mismatch between co-evolving domains. Eventually	Bassanini and Giovanni (2001); Burgelman and Grove (1996); Castaldi and Dosi (2005); Coriat and Dosi (1998);

	this mismatch triggers change to unlock paths. For example, the routine for capacity utilization of manufacturing equipment in the Intel case was originally developed to guarantee a demand-driven production. But as the external environment changed, declining memory prices and increasing competition, the same routine was responsible for the initiation of a company-wide transformation process (Burgelman & Grove, 1996).	March (2006); Martin and Sunley (2010); Sydow, et al. (2009)
Imperfect adaptation	Rule guided behavior or the performative side of routines are never fully predictable but exhibit at least some variation in practicing rules or routines (Becker, 2006). This kind of variation induces flexibility into the organization and facilitates change processes eventually leading to the unlocking of an organizational path.	Bassanini and Giovanni (2001); Pentland, et al. (2012)
Heterogeneity	Heterogeneity found in the beliefs of, for example, agents, corporate strategies, assigned job roles, technological know-how, individual or group behavior, preferences, or organizational structures have a tremendous impact on the ability of an organization to effectively unlock paths. Actually heterogeneity may be observed as a meta mean for unlocking paths as without heterogeneity unlocking is according to the cybernetics law of variety impossible (Ashby, 1956).	Ashby (1956); Bassanini and Dosi (2000); Castaldi and Dosi (2005)
Reallocation of resources	If an organization and the individuals within this organization possess excessive resources, reallocation of these resources can overcome persistence and induces change that eventually unlocks paths. Such uncommitted resources, named “slack resources”, may evolve in organizations during times of success. In a crisis, these resources can be assimilated or reallocated and enable an organization to unlock paths. For example, in a study on airlines Cheng and Kesner (1997) found that an increase in slack resources to enhance external market effectiveness also increases the extent to which airlines respond to environmental shifts.	Sydow, et al. (2005); Bourgeois III (1981); Cheng and Kesner (1997); Nohria and Gulati (1996)

These means to unlock paths are similar in that they make use of or induce diversity into the organization. The term heterogeneity is casually used in management literature and sometimes referred to as variety, flexibility, differences, or diversity. Heterogeneity can be defined as "*the distribution of differences among the members of a unit with respect to a common attribute*" (D.A. Harrison & Klein, 2007: 1200). Sources of heterogeneity are discussed very broadly in literature addressing gender, race, age, tenure, education, functional background, marital status, cognitive structures, prior experiences, attitudes, individual performance, affect, or network ties (Bingham, et al., 2007; D. A. Harrison & Klein, 2007; Page, 2007). A drawback of current analytical models incorporating path dependence is that they do not allow for or explain the evolution of heterogeneity in organizations. For example, Arthur's (1989) Pólya urn model of path dependence assumes that in the lock-in phase, heterogeneity is absent and therefore the adoption pattern is reproduced infinitely. After lock-in occurred, the number of balls from the color not locked in is so small that the probability of drawing this color tends to zero. Similarly, organizational learning models incorporating path dependence do not allow for deviant belief sets in the lock-in phase (K. D. Miller, et al., 2006). Here all agents in the organization reproduce one set of beliefs infinitely in the stable equilibrium. But this is in stark contrast to empirical observations that organizations exhibit heterogeneity and variation in behavior, even in the lock-in phase (Sydow, et al., 2009). Therefore, an extended model of organizational path dependence by building upon the model of Sydow et al. (2005) is proposed. In accordance with the notion of punctuated equilibrium an exogenous environmental shock may cause unlocking by the reallocation of heterogeneous resources or beliefs present in the lock-in phase. After unlocking has occurred, the process of path dependence may be experienced again, reflecting the properties of a punctuated equilibrium process. Figure 6 shows the proposed four-phase model of path formation and unlocking of Sydow et al. (2005). The first three phases describe the formation process of path dependence as already outlined in the three-phase framework. With unlocking, a fourth phase is added to the basic framework of organizational path dependence. But while Sydow et al. (2005) added the fourth phase, to explain the case of unlocking, the constituting exogenous shock needed to unlock paths is not included. As already discussed, at the end of the lock-in phase an exogenous shock is needed to trigger the unlocking of paths. After the unlocking, the self-reinforcing process may repeat itself and start over from *Phase 1* of the formation process. Now the framework reflects a continuously and dynamic process of path formation and unlocking.

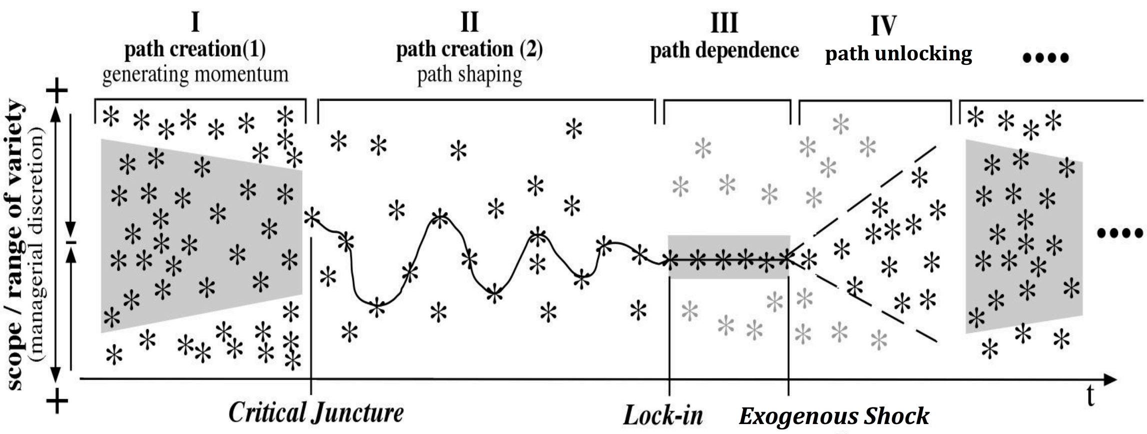


Figure 6: Adapted four-phase model of path dependence (Sydow, et al., 2005: 32)

As these remarks are up to now only presumptions about how unlocking in organizations could occur, evidence is needed in order to credibly claim the validity of the model. Although means for unlocking paths have already been proposed, none of them were tested as prior research efforts mostly focused on the first three phases (Castaldi & Dosi, 2005; Sydow, et al., 2009). Sydow et al. (2005: 22) point this out by stating that:

"It should be emphasized that none of these approaches deals explicitly with issues of path dependencies, not to mention provides a theory of unlocking paths that builds upon a theory of path constitution and specifies the conditions under which a once chosen path may be unlocked."

To prove that a process of path dependence can be unlocked, with the definition of path dependence being a process governed by increasing-returns, contingency and lock-in, needs theoretical clarification. So, assuming that the four-phase model describes path dependence and unlocking correctly, unlocking must be possible after the lock-in phase. Furthermore, unlocking must be possible with one of the aforementioned means that induce or preserve heterogeneity. The question is how the logic of unlocking can be integrated in the four-phase model. To answer this question a method correctly capturing the path formation process and giving the possibility to unlock is needed. As a starting point, different methods applied so far in organizational research on path dependence are reviewed and discussed in order to find a suitable methodological approach.

3. Method for Examining the Unlocking of Paths

Prior studies on path dependence made use of a broad set of methods. These methods include stochastic models (Arthur, 1989), historical narratives (David, 1985; Mahoney, 2000), qualitative case studies (Gavetti & Levinthal, 2000; Holtmann, 2008), controlled laboratory experiments (Koch, et al., 2009; Langer, 2011), structural equation models (Mallach, 2013), quantitative longitudinal studies (Schulte, 2013), computer simulations (Meyer, 2012; Petermann, et al., 2012; Seidel, 2012), and purely theoretical approaches (Pierson, 2000; Sydow, et al., 2009). The variety of methodological approaches seems to reflect the manifold definitions of path dependence.[17] These inconsistent definitions lead to a discourse on how to test for properties of path dependence (Dobusch & Kapeller, 2012; Garud, et al., 2010; Vergne & Durand, 2010). The present discussion is primarily about shortcomings of specific methods, when examining path dependence, and is split broadly into two groups. One, which emphasizes the advantages of qualitative approaches, like case studies for explorative theory building (Dobusch & Kapeller, 2012; Garud, et al., 2010), and another, which proposes controlled quantitative approaches, as for example computer simulations or laboratory experiments, for theory testing (Castaldi & Dosi, 2005; Vergne, 2013; Vergne & Durand, 2010). To overcome the methodological dispute, a clear definition of path dependence may prove helpful and give guidance in finding an appropriate research design.

3.1 Methodological Issues in Path Research

As previously mentioned, path dependence is defined "*as a property of a stochastic process which obtains under two conditions (contingency and self-reinforcement) and causes lock-in in the absence of exogenous shock*" (Vergne &

[17] In social science the definition of path dependence is not consistent. For example, different definitions of path dependence have been given by Mahoney (2000), Nelson and Winter (1982), Sydow, et al. (2009), Vergne and Durand (2010), Arthur (1989), David (2001) or Pentland, et al. (2012).

Durand, 2010: 737). The methodological consequences of this definition are to be discussed subsequently in detail. In particular, it will be elaborated how emergence, contingency, and lock-in affect the choice of a proper research design. Furthermore, emphasis will be put on the methodological difficulties in observing the unfolding of historical processes.

3.1.1 Problems in Observing a Historical Process

Since it is often only possible to observe one specific history at a time, making propositions about how, and sometimes even if, history matters is seriously hampered from an empirical point of view (Vergne & Durand, 2010). More precisely, in order to ex-post attribute the outcome of a process to stochastic events in history, students need to observe how the development of the process would have changed under different initial conditions and different historical events (Castaldi & Dosi, 2005).[18] Only then it can be ensured that contingent events, that were identified as drivers for path dependence are responsible for the outcome. Else, it could be argued, that the outcome of the process was already determined by initial conditions or that different contingent events would have lead to the same result. But repeating a real world process with different histories ex-post and examining what would have changed is not feasible, therefore, it can not be assured if and how much history mattered (Gould, 1977). For example, with regard to the QWERTY narrative, it may be postulated whether the mentioned typewriter contests were crucial events for QWERTY to win the race, or if other events were more decisive (Liebowitz & Margolis, 2013). It even may be argued that the events were not decisive at all and QWERTY would always win, when rerunning the tape of history (Kay, 2013). In retrospect, one may only suggest or assume that some events were decisive. Yet, it cannot be affirmed with absolute certainty, if the events at hand are the ones driving the path formation process or if other, perhaps less obvious, events were more important. It remains ambiguous, if the history as perceived today, through narratives and case studies, actually reflect decisive

[18] For an explanation why initial conditions and historical events have to differ in order to compare processes, see chapter two for remarks on non-ergodicity.

events, because we do not know what would have changed when the events played out differently (Vergne & Durand, 2010). Therefore, ex-post conducted research, such as case studies, may not rule out the possibility that other events were responsible for the outcome, or that history could not have been played out differently.

In order to overcome the problems in ex-post analysis of paths, one could suggest observing history as it unfolds. Ethnographic approaches, like social anthropology, are known for yielding rich longitudinal data, and may prove helpful in examining unfolding processes over time (Van Maanen, 1979). But again, the importance of small events may not draw the attention of the researcher, because of individual bias' or incorrect interpretations (Castaldi & Dosi, 2005). Also, it is only possible to observe one history, and not multiple histories, at a time, so the decisiveness of the contingent events may again be questioned. Furthermore, the feasibility of empirically observing paths in the making may be questioned, because of the long time frames involved. Take for example the amount of time it took QWERTY to achieve a stable equilibrium (David, 1985), or until an exogenous shock disrupts a company like the Bertelsmann Book Club (Holtmann, 2008). A solution to this dilemma is to use methods, which allow for taking control over history and are classified as "history friendly" (Malerba, et al., 2008). Instead of identifying crucial events as time passes, history may be artificially constructed by inducing these events into a controlled environment. Experiments, for example, provide researchers with the possibility to apply different treatments to groups and compare the results of groups having received a treatment to those who did not (Webster & Sell, 2007). By using an experimental design, the impact of history may therefore be studied while controlling for initial conditions at the same time (Koch, et al., 2009; Langer, 2011). When examining and comparing different historical processes, experimental research designs have an advantage over ex-post conducted studies or ethnographic approaches. It is therefore advisable to make use of these "history friendly" methods, when examining the formation of paths. Besides the difficulties in proving that some events were decisive for the outcome of the process, the need to show that these events are also contingent proves to be difficult as well.

3.1.2 Problems in Validating Contingency

The necessary condition of contingency, as a property of path dependence, puts the examination of organizational paths to a hard test. From a mathematical perspective, empirically validating contingency or randomness is impossible (Gödel, 1931; Vergne & Durand, 2010). Thus, empirical studies are challenged to falsify the contingency condition by attributing non-random patterns to the sequence of events that are responsible for the process outcome. Nevertheless, even if such patterns are not detected, the process may still be non-contingent, as there readily could be information unobserved by the researcher, perhaps too inconspicuous to detect. Simply escaping the dilemma by removing contingency as a necessary condition for path dependence would affect the theoretical core of the concept. In order to differentiate path dependence from similar concepts this must be avoided by all means (Kuhn, 1962; Sydow, et al., 2009). Therefore, a research design that at least allows reasonable assumption on the presence of contingency, or at least pseudo-randomness, needs to be chosen. This again demands a controlled environment, were it is possible to randomly induce events and examine the outcome in comparison to other events. So instead of empirical proving contingency, the research methodology itself allows to include randomness. Experiments (Koch, et al., 2009) or computer simulations (Seidel, 2012) in research on organizational paths induce events, which may be considered random.

3.1.3 Problems in Examining Emergence

Individuals in organizations are socially embedded and interact with each other (Orlikowski & Yates, 1992). Interaction between individuals occurs, for example, through calls, informal chats, meetings, or e-mails. Since these interactions result in interdependencies between individuals and therefore a complex system, unanticipated properties may "*emerge*". Emergence can be defined according to Stacey (1996: 287) as:

"...the production of global patterns of behavior by agents in a complex system interacting according to their own local rules of behavior, without intending the global patterns of behavior that come about. In emergence, global patterns cannot be predicted from the local rules of behavior that produce them. To put it another way, global patterns cannot be reduced to individual behavior."

Emergence is an important property of path dependence, as paths *"often unfold behind the backs of the actors"* (Sydow, et al., 2011: 322). Yet, analytical research methods in social science have difficulties in bridging the gap between individual interactions and the emergent organizational behavior (Goldspink & Kay, 2004). Multi-level analysis and theories often apply individual concepts to higher levels, without taking into account the specific problems of aggregation (Felin & Foss, 2005). Furthermore, conventional statistical methods assume a cause-effect relationship and do not account for nonlinearities in interactions between variables. Therefore, conventional statistical methods are insufficient to examine emergence (M. Schneider & Somers, 2006). To overcome the limitations of traditional methods in studying complex systems with emergent properties, once again, experimental methods prove useful (H. Arrow, et al., 2000). Further on, researchers from diverse fields (such as physics, genetics, politics, or economics) highlight the advantages of computer simulations in studying complex systems with emergent properties (Axelrod, 1997b; J. R. Harrison, et al., 2007). As simulations allow the observation of multiple interactions over long time frames, emergent properties can be traced back to the individual level; or as Castaldi & Dosi (2006: 108) put it, *"from micro behaviors to system dynamics, and back".* Therefore, for examining and tracing emergence, simulations prove to be an effective method.

3.1.4 Problem of Lock-In Identification

In the extended Polyá urn of Arthur (1989) lock-in is simply defined as an infinitely repeated stable equilibrium state. Once a technology is locked-in, agents decide to adopt only the locked-in technology and reject other available technologies. But, while the binary definition of a lock-in holds in formal models, the application of this definition on processes in the real world may be challenging. Nowadays, no one would claim that consumers are locked into VCR recorders or personal computers with 640k memory restrictions. This raises the question of whether consumers, or an economy, were truly locked into these technologies, or if it just was a period of meta-stable equilibrium leading to a superior technology in the long-term. But, although time is a crucial component of lock-in, it has not yet received much attention (for an exception see Vergne & Durand, 2010). Pragmatically, prior empirical studies suggest that a lock-in has occurred if the equilibrium persists over a “long” time frame, but have not defined what long means. For instance, the exact time point when QWERTY achieved a stable equilibrium, and even if the equilibrium is stable, remains unclear (David, 1985, 2007). Again, to overcome the difficulties of defining lock-in, experiments are of great help, as they may allow proving that the underlying patterns are reproduced, even if the potential for choosing another solution is given. For example, Koch et al. (2009) examine the impact of complexity in the individual decision-making process on lock-in through an experimental study, and show that individuals do not switch to superior solutions. Virtual experiments, by means of computer simulations, are also very precise in showing that lock-in occurred. As the state of a computer simulation can be observed at any time step, a lock-in can be defined by showing that the system does not change over time.

In conclusion, for a historical process with contingent events, emergence, and lock-in, an experimental research design is suitable. Also, as previously argued, most qualitative and quantitative methods are not able to capture these properties altogether. Because of this, experimental studies are particularly suitable for examining path dependence.[19] Up till now, only a methodological recommendation was given to use an experimental method, such as a laboratory experiment or a computer simulation experiments. Therefore, the specific requirements for an experiment in light of the research problem are derived. As the research objective is to develop a model of organizational path dependence, including the logic of unlocking, the design has to allow for examining path formation, and subsequently the unlocking of accrued paths in organizations. To depict these processes, an experimental research design has to take care of the following characteristics:

- First of all, the design must be capable of capturing the essence of an organization as a social system. Organizations are complex systems, consisting of individuals, which interact according to rules in order to achieve their goals (March, 1981; Simon, 1964). Organizations are viewed to adapt to their environment through such rule-guided behavior (B. Levitt & March, 1988). An organizational research design must therefore include interactions between individuals and provide an environmental context.

- Second, the nonlinearities inherent in the self-reinforcing mechanisms of path dependence have to be traced back to the behavior of individuals within the organization. An experimental study must explain how the micro level of an organization, here individual actors and the interactions between these individuals, is influencing the emergent behavior of an organization (Castaldi & Dosi, 2005).

- Third, a longitudinal research design is required for examining the three consecutive phases of path dependence and the additional unlocking phase. The four phases must be clearly distinguished, which makes it necessary to measure the current state of the system in the course of the process at any time.

[19] Nevertheless it has to be highlighted that using a different definition of path dependence other research approaches may also be recommendable.

- At last, the research design must include the previously mentioned properties contingency, emergence, and lock-in.

Reconstructing such a setting in a laboratory experiment could yield interesting findings. In particular, because results obtained through an laboratory experiment may be generalized and extrapolated to a world outside the lab (S. D. Levitt & List, 2007). Also, laboratory experiments are well suited for examining path dependence at the individual or small group level (Vergne & Durand, 2010; Webster & Sell, 2007). But for studying organizational paths and the unlocking of organizational paths, laboratory experiments prove impractical. Including and controlling for all the previously mentioned properties is a too ambitious undertaking. For example, selecting a sufficient number of suitable participants with heterogeneous backgrounds found in real organizations, and afterwards observing the nonlinearities in their behavior, is hardly possible. Also, the complexities arising in organizations are hard to examine within a laboratory experiment (Zelditch, 1969). Computer simulation experiments offer a remedy by being capable to mirror organizations virtually. Therefore, simulations are of great help, when examining complex systems, like organizations, and their emerging properties over time (Lant & Mezias, 1992). Virtual experiments also allow for tracing the self-reinforcing mechanisms that lead to lock-in, observing and comparing multiple historical trajectories, and are especially suitable to examine the probability of lock-in (Vergne & Durand, 2010).

Because of this, the dissertation applies the simulation methodology in order to answer the research questions. Before the state of the art in simulation research is reviewed and simulation experiments are specified, a proper introduction into the "art of simulation" (Axelrod, 1997a) will be given by delineating a simulation protocol, comparing different simulation methodologies, and mentioning examples of computer simulations in path research.

3.2 Artificial Organizations: The Simulation Methodology

In a nutshell, computer simulations can be comprehended as virtual worlds, populated with artificial agents, that are behaving according to predefined fixed laws (J. R. Harrison, et al., 2007). Generally, these laws are formalized by, (a) parameters expressing the initial state of the model and, (b) parameters defining the transition probabilities between different states. Based on the assumptions made in the virtual worlds, experiments are conducted by varying the parameters of the model and then measuring changes in the outcome (Carley & Newell, 1994). New simulation frameworks and powerful computing technologies have lead to a rapid increase of simulation studies in the field of social science over the past two decades (Axelrod, 1997a; Ganco & Hoetker, 2009; Richiardi, et al., 2006). Some seminal findings have been achieved through computer simulations, as, for example, the tradeoff between exploration and exploitation organizations face (March, 1991), or the way organizations search for solutions to problems (Levinthal, 1997). Still, the method relegates a nice existence in leading journals, when compared to large-scale empirical or qualitative case studies. Particularly, in the field of management and organization research, computer simulations have a negligible impact beyond specialized simulation journals, potentially inhibiting dissemination to a broader audience (J. R. Harrison, et al., 2007). As possible explanations for this pitiful state, the lack of a common simulation protocol in social science (Richiardi, et al., 2006), limited methodological and philosophical understanding on the side of management scholars, as well as insufficient training in computational modeling have been identified (Davis, et al., 2007; N. Gilbert & Terna, 2000; J. R. Harrison, et al., 2007). While providing an extensive training is out of scope for the work at hand, enabling social scientists to understand the development, procedure, and results of computer simulations is feasible. Therefore, the goal of this section is to briefly classify the simulation into a broad philosophical and methodological framework, and to provide a standardized simulation protocol as a common theme for conducting simulation research.

3.2.1 Methodological and Philosophical Issues of Simulations

The philosophy of research broadly distinguishes between two mindsets in scientific reasoning: deduction and induction. In deductive reasoning, bundles of hypothesis are postulated, based on assumptions about a phenomenon. These assumptions may then be translated into mathematical relationships. Afterwards, the hypothesis are empirically tested and then confirmed or rejected. A problem with deduction is, that social processes might be too complex for mathematical derivation (J. R. Harrison, et al., 2007). Conversely, inductive reasoning derives from empirical observations, explanations about generalizable relationships in order to build theory. A problem with empirical studies is, that data is difficult to obtain, because variables might be unobservable or difficult to measure (J. R. Harrison, et al., 2007). Classifying computer simulations either as deductive or inductive would neglect the peculiarities of the method. This is why simulations are often considered to be a “third way of doing science” (Axelrod, 1997b). In fact, computer simulations share similarities with both scientific approaches. Like in deductive reasoning, a computer simulation has to start with assumptions about a real world phenomenon. Based on these assumptions, a model is derived and transferred into computer code. Executing the computer code generates data from which conclusions about general relationships may be drawn by applying inductive methods. Instead of empirical data from the real world, the underlying rules in the simulation provide rich data about the consequences of the models assumption (Axelrod, 2007). By this, simulations eliminate the shortcomings in the deductive analytical models’ inability to capture the complexity of the real world and the difficulties in acquiring empirical data of inductive research. Simulations are therefore especially suitable when deductive or inductive approaches are not feasible or very difficult to apply (Carley, 1995). The dissertation follows the “third way of doing science” by proceeding in accordance with a simulation protocol, which shows elements of deduction and induction.

3.2.2 A Protocol for Simulation Research

For qualitative methods, such as case studies (Yin, 2009), quantitative methods, like structural equation modeling (Schumacker & Lomax, 2010), and mixed methods (Jick, 1979), extensive literature with guidelines on research designs is readily available. The situation is quite different for simulation research, as there is no common standard guideline for conducting simulations. Social scientists counter that the freedom the simulation methodology provides may lead to a state of anarchy (Richiardi, et al., 2006). As a remedy for this situation, frameworks, protocols, and stage models to conduct simulation research have been proposed in order to provide a more structured approach to social simulations (Axelrod, 1997a; Davis, et al., 2007; N. Gilbert & Troitzsch, 2005; J. R. Harrison, et al., 2007; Lorscheid, et al., 2011; Polhill, et al., 2008; Richiardi, et al., 2006). Capturing the essence of these different frameworks, a six-stage approach to simulation research will be described, which serves as protocol for deriving a simulation model and conducting virtual experiments.

Stage 1: Methodological Fit of the Simulation

The starting point of every simulation research project has to be an intriguing research question, that suites the computer simulation methodology (N. Gilbert & Troitzsch, 2005). Simulations prove especially useful for research problems comprising of processes unfolding over long time frames and spanning over several observation levels (Davis, et al., 2007). If, in addition, the complexity of the object under analysis does not permit a closed form mathematical solution, and data is hard to obtain, the simulation method should be favored (Davis, et al., 2007; J. R. Harrison, et al., 2007). Path dependence emerges in organizations from interactions between individuals, potentially unfolding over a longer period and, because of contingent events in the process, data is hard to obtain. Furthermore, obtaining data on the effects of different means to break paths is challenging. With concern to the state of theory building, simulations should be applied in a premature research stage, where simple theory already is in place, but understanding of concepts is still limited (Davis, et al., 2007). While it may be argued, that the variety of empirical studies on path dependence already gives a good understanding of path formation, this is not true for the case of unlocking paths (Ericson & Lundin, 2013). As set out in Chapter 2, the logic of path dependence (Chapter 2.1 and Chapter 2.2) and first means for unlocking paths (Chapter 2.3) have been proposed, but theory is still porous and undeveloped. Furthermore, Chapter 3 provides an overview why experiments, and in particular virtual experiments, are suited for path research. Therefore, the simulation method seems to be a good fit for the research objective.

Stage 2: Development of the Simulation Model

The development of a simulation model comprises two tasks: first, the selection of a specific simulation approach and second, the derivation of a formal model. Davis et al. (2007) compares the selection of the simulation approach to choosing a theoretical framework, because a simulation approach implicitly makes theoretical assumptions about the object of interest. Hence, the selection of a simulation approach is always accompanied by underlying theoretical constructs. In order to build a formal model, preliminary considerations should be made with regard to the simulation model type. Chapter 4.1 and Chapter 4.2 will describe and compare simulation frameworks used in management research.

Following this, a formal model capturing the most important assumptions about the phenomenon must be derived. Models are an abstract representation of the reality and used to explain ongoing processes in the world (Lave & March, 1975). Usually, models exhibit a mathematical form with systems of equations, serving as rules on how actors or organizations behave and interact. Processes in organizational models must at least comprise of individuals, social processes between these individuals, and an organizational structure connecting these individuals (Van Horn, 1971). In the case of simulations, the formal model also depends on the assumptions of the chosen simulation approach. The development of a formal model is subject of Chapter 5.

Stage 3: Transferring the Formal Model into Computer Code

After deriving the formal model, it must be transferred into machine-readable code. Chapter 5.7 is dedicated to the selection of a framework for transferring the model into code, explaining the characteristics making up good computer code, and highlighting the peculiarities of writing computer programs.

Stage 4: Conducting Virtual Experiments

The design of the virtual experiment has to be specified before it is performed, through altering parameters in the computer model. The experimental design consists of five items: initial conditions, time structure of the simulation, outcome measurements, number of iterations, and the number and range of variations (N. Gilbert & Troitzsch, 2005; J. R. Harrison, et al., 2007). While the transition from one state to another is defined by the rules in the model, initial conditions need to be specified before running the simulation. Afterwards, the length of the simulation, measured in number of steps, needs to be computed. Generally, a stable equilibrium state is used as stopping criteria (Axtell, et al., 1996). At the end of each time step or simulation run, the outcome of the current model state is measured and stored. To capture the behavior of the system, multiple iterations of the simulation are necessary, because random elements in the simulation make single runs not representative for the model (N. Gilbert & Troitzsch, 2005). The number of iterations therefore has to be computed. To examine the influence of parameter variations on the simulation results, all parameters are assigned a value range. Typically, the design of a computer simulation is supported by robustness analysis to trace the effects of variations on the outcome (Chattoe, et al., 2000). Chapter 5 includes the parameter variation and discusses the chosen parameters.

Using the experimental design, the virtual experiments are then carried out, by repeating the simulation with different parameter settings. Variations in the parameters allow for statements with regard to the behavior of the model and are useful to build new theory. Besides varying the input parameters, experiments are conducted by unpacking a construct into subsequent constructs, varying the assumptions of the model, or adding new features to the simulation model (Davis, et al., 2007). In Chapter 6, virtual experiments are derived, the simulation model is extended, and outcomes are measured. After the virtual experiments were conducted, the results must be prepared, analyzed, and discussed.

Stage 5: Analysis and Discussion of the Results

By preparing data, the task of translating the measurement outcomes of a simulation run into a form allowing for analyzing the data is meant. While some simulation tools prepare the data automatically, usually the researcher faces an extensive output, such as one written in an unformatted text file. The stored data then has to be transferred into a statistic or spreadsheet program like SPSS, Microsoft Excel, or Apple Numbers, where it can be further analyzed.

In general, the output data of a simulation model may be analyzed in the same way as empirical data, but attention has to be drawn as simulations mostly exhibit non-linear relationships (J. R. Harrison, et al., 2007). A popular mean for preparing and analyzing large amounts of simulation data is by using two- or three-dimensional graphs (N. Gilbert & Troitzsch, 2005). With graphs, the relationship between different parameters may be easily revealed and illustrated, or the behavior of a simulation model over time is examined. By uncovering formerly unknown effects or relationships, the results may contribute to theory building.

Stage 6: Validation of Results

In a final step, the findings of the simulation study should be confirmed by collecting empirical evidence. The outcomes may guide researchers to new strategies in obtaining empirical data, or hint to formerly unknown relationships, that can now be tested systemically. As the empirical testing of the simulation is beyond the purpose of the dissertation, further studies have to confirm the results of the simulation model. Nevertheless, Chapter 7 provides limitations of the study and advice for further research. Figure 7 connects the structure of the thesis to the six stages of the simulation protocol.

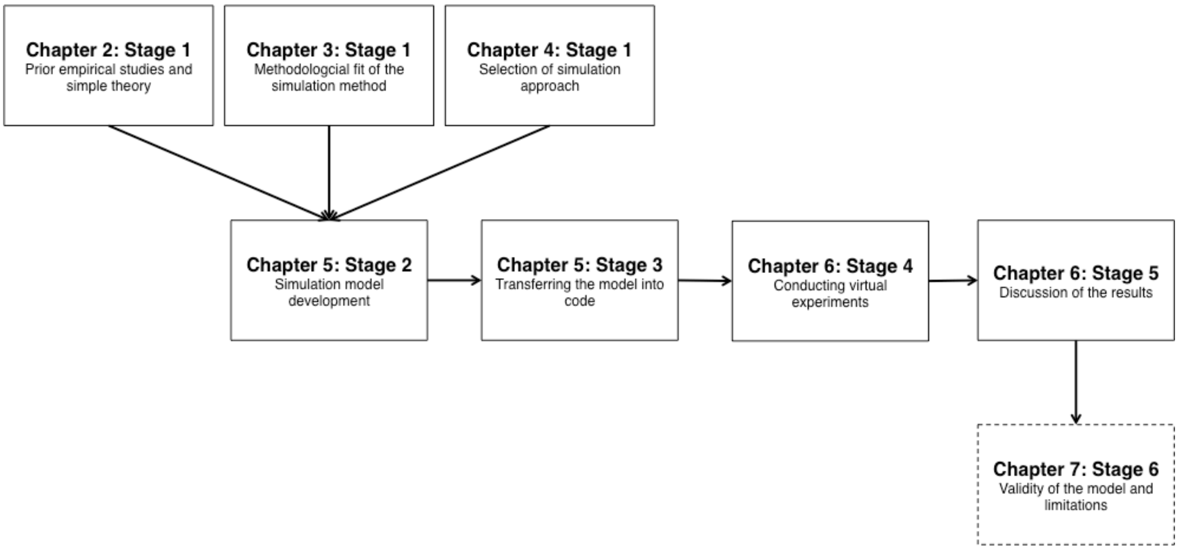


Figure 7: Thesis chapters along the stage protocol for simulation studies

4. State of the Art in Simulations

The history of simulation dates back to World War II, when it was first applied by John von Neumann and Stanislaw Ulam in the Manhattan Project to understand the behavior of neutrinos (Casti, 1996). Although some important modeling insights were achieved back then, the field of social science adapted the method only in the 1970s (for examples see Schelling (1971) or M. D. Cohen, et al. (1972)). Despite the early interest in simulations in the field of social science, the method is still new to most of the scholars in management research. In the 1990s, some seminal contributions were achieved in management research with help of simulations and the number of publications increased (for examples see March (1991), Levinthal (1997) or Lant and Mezias (1992)). But, publications still lagged behind in comparison to empirical studies (J. R. Harrison, et al., 2007). And while there are still researchers suspicious about the method, simulations gained a permanent place in the field of management research.

4.1 Simulations in Management Research

The decision for a specific simulation approach is closely related to choosing a theoretical framework (Davis, et al., 2007). The type of simulation constrains, but at the same time qualifies, the researcher in exploring new theory. Commonly used computer simulation approaches in management research, and particularly in path dependence research, include system dynamics and agent based models, like NK-models or cellular automata (J. R. Harrison, et al., 2007).[20] Each of the simulation approaches has its advantages and disadvantages. Therefore, in the following, an overview will be provided for identifying a suitable simulation approach.

[20] While, in general, one can think of further approaches, such as genetic algorithms, the discussion will be restricted to the most common simulation models (Davis, et al., 2007; J. R. Harrison, et al., 2007).

4.1.1 System Dynamics

System dynamics models are used to describe the structure and behavior of a complex system as a whole, instead of modeling each part of the system on its own (Forrester, 1958). In system dynamics models, the interactions between individuals or units are usually formalized in a system of differential and difference equations, describing the current behavior. Furthermore, through rules in the model, future states of the system are predicted by the simulation (Forrester, 1980; N. Gilbert & Troitzsch, 2005). For that reason, system dynamics models are often used as “management flight simulators” (J. Sterman, et al., 2013). They allow investigating the consequences of different policies in a controlled virtual environment, before actually implementing it within the organization (J. D. Sterman, 2000). Because of these properties, system dynamics models are not only used by social scientists for theory building, but also applied by practitioners in consulting firms and companies to assist the decision making process of management teams. Furthermore, the focus of the research interest lies often in the influence of initial conditions and input factors on the outcome of a process (Davis, et al., 2007).

System dynamics models are constituted using a standardized modeling language, comprised of symbols known from thermodynamics. In particular, a system dynamic model is fully defined by an initial supply of objects, a time-dependent valve regulating the flow from the initial supply into inventories, flows between different inventories, and positive or negative feedback loops (Figure 8b). For a better understanding, the previously discussed generalized Pólya urn process is shown in a system dynamics stock and flow diagram with causal feedback loops in Figure 8a (Arthur, 1989; J. D. Sterman, 2000). The two bold arrows, at the top and bottom of the graph, symbolize flows from an inventory of white and black stones, regulated by a valve into a stock of stones. At each time step, the positive feedback loop R determines the order rate of the valve, defining the probability of drawing a white or black stone. If all stones from the inventory are exhausted, the simulation run is terminated and the end state of the system is observed.

(a) (b)

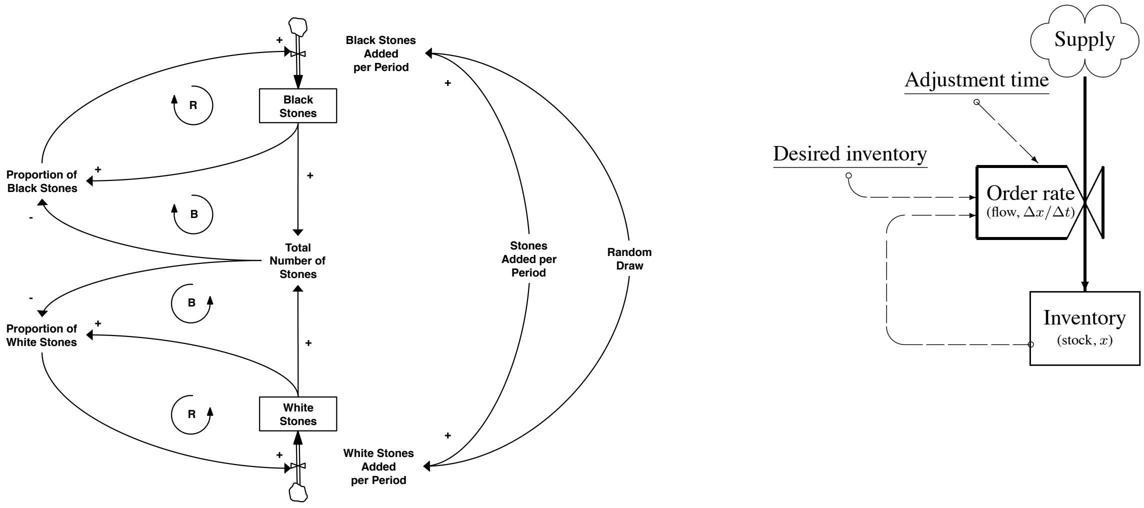


Figure 8: (a) Stock-flow diagram of the Pólya urn process (see Sterman, 2000: 355) and, (b) legend of system dynamics symbols (N. Gilbert & Troitzsch, 2005: 30)

Evolving around the system dynamics community, a variety of models have been developed for the purpose of theory building. With regard to path dependence, models examined the dynamics of scientific revolutions (Kuhn, 1962; J. D. Sterman & Wittenberg, 1999), network effects in the video recorder industry (J. D. Sterman, 2000), dynamics of complex technology markets (Schwaninger & Mandl, 2012), distinguishable forms of path dependence respectively path independence in the insurance industry (Mandal, 2001), and propositions about tipping points in the path formation process (Bramson, 2008). Nevertheless, a major limitation of using the system dynamics approach in examining organizational paths is the restriction to the macro level. Instead of modeling how properties emerge from interactions, the emergent property itself is modeled (N. Gilbert & Troitzsch, 2005).[21] But, considering that organizational paths emerge from interactions between individuals in an organization, the lack of simulating the micro-macro links is a significant limitation (Castaldi & Dosi, 2005). Therefore, a simulation approach

[21] Schieritz & Milling (2003) draw an analogy to a forest: agent-based simulation is modeling the trees and examine, how they make up a forest, while system dynamics simulations model the structure of the forest as a whole. Regarding the concept of path dependence see also Meyer (2012).

is needed, which is capable of simulating a level made up of interacting individuals and observing the emergent effects, resulting from these interactions on an aggregate level. Recently, agent based models became popular for examining such emerging phenomena (Macy & Willer, 2002).

4.1.2 Agent Based Simulations

Traditional methods in social science may encounter problems in linking the behavior of individuals to the big picture of social structures (Ellis, 1999). The debate on the micro-macro divide, has a long standing history in social science (Alexander & Giesen, 1987). To bridge this gap, it has been proposed to use agent-based models for depicting the interactions between actors in an organization, and observing the emergent behavior of the social system (N. Gilbert, 1995; Macy & Willer, 2002; Sawyer, 2003). In contrast to system dynamics models, the aim of an agent based model is therefore not the prediction of future system states, but emphasis is put on the explanation and understanding of emergent behavior on system level (Billari, et al., 2006). Likewise, and different to micro simulations, agent based models are made up of heterogeneous agents, acting autonomously based on their beliefs, cognition, or knowledge and taking actions according to a predefined set of behavioral rules (Conte, et al., 2001). More precisely, agents are not controlled by other agents, exchange information with peers on basis of a common language, react to perceived changes in their environment, and actively take actions to reach their goals (Woolridge & Jennings, 1995). To meet these prerequisites, agent based models mostly include multiple agents, social structures with rules defining the communication between agents, and an exogenous environment influencing the decisions made by agents in the system (N. Gilbert & Troitzsch, 2005). Since agent based models are able to observe emergence and explain it through the dynamics on the micro level, the method was quickly adopted by researchers of organizational path dependence (Vergne & Durand, 2010). For instance, agent based models are used in path research to examine the assertion of self-reinforcing effects in hierarchical organizations (Petermann, et al., 2012), diffusion processes in two sided technology markets (Meyer, 2012), organizational adaptation through learning (March, 1991), the influence of complexity and environmental change on organizational path dependence (Seidel, 2012), or the diffusion of open source business software (Bonaccorsi & Rossi, 2003).

Within the agent based model approach, different types of frameworks are further distinguished. Apart from highly individualized models, using generic algorithms, cellular automata and NK models are particularly relevant for organizational research (N. Gilbert & Terna, 2000; Hegselmann, 1996). Strictly speaking, cellular automata are an antecedent to the modern agent based models (Morand, et al., 2010; Schelling, 1971). Subsequently, both frameworks will be briefly described, have their underlying characteristics carved out, and the relevance for researching organizational paths will also be discussed.

Cellular Automata

Cellular automata represent a separate class in the field of agent based models, as interactions are assumed to be local and the spatial dimension remains fixed over the duration of a simulation run (Brandte, 2007; J. R. Harrison, et al., 2007). Broadly speaking, cellular automata consist of multiple cells with several possible states, arranged in a lattice or grid. The state of each cell itself depends on a set of rules and on the state of neighboring cells. Usually, the grid is a two-dimensional rectangle, however, it also can be three-dimensional or triangular. Common neighborhood definitions in a two-dimensional cellular automata are based on the work of von Neumann and Moore (Hegselmann, 1996; von Neumann, 1966).[22] Figure 9 describes the two different neighborhood typologies and possible interaction patterns. The local neighborhood definition takes a very simplistic approach on how individuals interact within an organization, compared to more

[22] In these neighbourhoods the bond of interactions is defined by the Manhattan distance parameter r.

complex constructs, found in studies on social network analysis (Knoke & Yang, 2008).

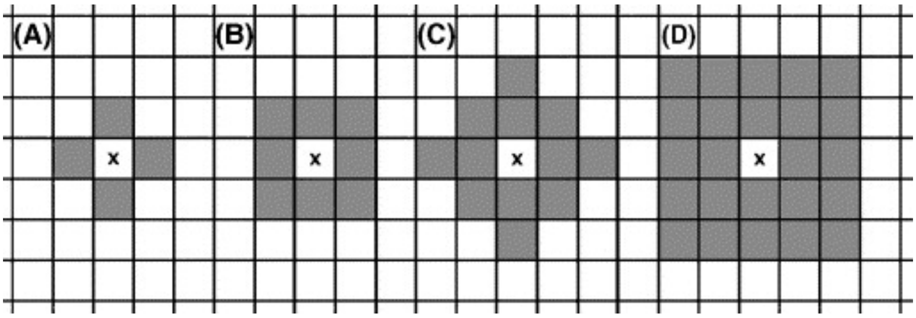


Figure 9: Typology of neighborhoods in cellular automata: (A) von Neumann neighborhood (r=1), (B) Moore neighborhood (r=1), (C) Extended von Neumann neighborhood (r=2), (D) Extended Moore neighborhood (r=2) (Fonstad, 2006: 221)

Because of the spatial interactions between cells, cellular automata are often used to examine emergent effects on macro level (N. Gilbert & Troitzsch, 2005). Literature in geographical economics, where location plays a major role, often make use of cellular automata to solve spatial problems with concern to land usage or economic development (Almeida, et al., 2008). Originating from this literature, extensive research on geographical path dependence was conducted. Examples include the investigation of the importance of geographic locations for path dependence in regional industry clusters (Brown, et al., 2005), the assertion of railway gauge tracks as technological standards in railway transportation systems (Puffert, 2002), and the historical agricultural development in western economies (Balmann, 1994). Apart from geographical economics, organizational simulation studies apply the cellular grid as a representation for firm structures and the bounded rationality of individuals in the field of management research (for examples see Brandte (2007), Lomi and Larsen (1996), K. D. Miller, et al. (2006), and K. D. Miller and Lin (2010)). By extending the grid representation with a tunable parameter for the interdependencies between interactions, the NK model is another approach to take into account the spatial dimension of organizations and path dependence in an agent-based model (Leydesdorff, 2002).

NK Models

The NK model was originally developed to solve problems evolving around the evolution of species and genes in biology, but recently became a widely applied simulation framework in management research (for examples see Baumann (2008), Ethiraj and Levinthal (2004), Ethiraj, et al. (2008), Gavetti (2005), Levinthal (1997), Rivkin (2000), Rivkin (2001) and Siggelkow and Rivkin (2009)). In particular, NK models are used to address coordination and optimization problems, arising out of interdependencies in decision making, organizational structure, or between product parts. Regularly, these problems are challenging to solve empirically (Ganco & Hoetker, 2009).

The NK model consists of one or more agents, searching for an optimal solution on a "performance landscape". A performance landscape consists of 2^N points, where every point has a specific performance value. Displaying the performance on the vertical axis would yield to a landscape, where the highest performance values form peaks. The combination of the letters NK in the model description stems from the two basic model parameters, defining the size (N) and ruggedness (K) of a performance landscape. More specifically, the parameter K defines the degree of interdependencies between the elements of a binary N-dimensional vector. To each element in the vector, a performance contribution value is randomly assigned, which depends on the performance of K other elements. Increasing K makes the performance landscape more rugged, as changing one dimension affects the performance of other dimensions. Agents are now supposed to search the performance landscape for the optimal solution, defined by the highest peak in the landscape. Furthermore, it is commonly assumed that agents only adapt new solutions, if they are superior to the current solution, and that the bounded rationality of agents restricts the search towards the local neighborhood.[23] One stopping criterion for the search is, that agents cannot improve their performance, because all of its neighbors exhibit a lower performance value. Combining complexity and local search may impede agents to find the highest performing solution in the landscape and leave them on a local peak in the landscape (Kauffman, 1993). To put it another way, the search process in a rugged landscape can be compared to a mountain hiker, aiming to reach the highest summit without using a map. By solely relying on his sight, the hiker may be ascending the nearest mountain, only to find out, that surrounding summits are even higher and he missed to climb the highest peak. Figure 10 exhibits an example for such a NK landscape.

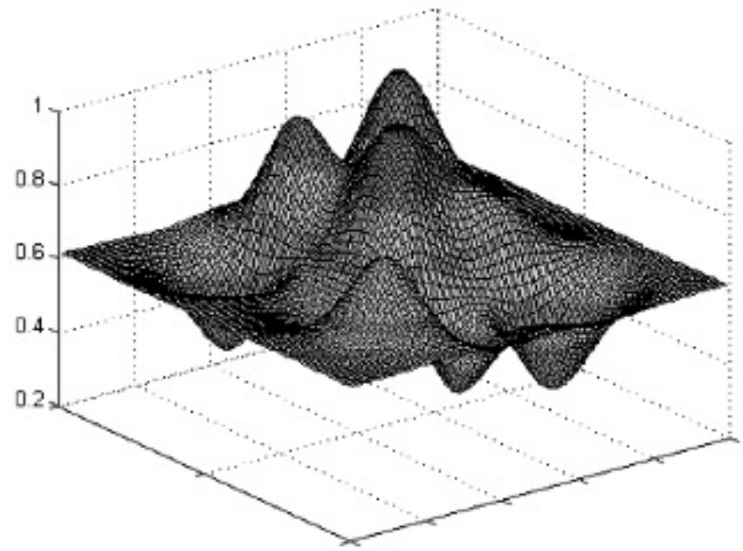


Figure 10: Example of a rugged NK performance landscape with three local optima and one global optimum (Caldart & Oliveira, 2010: 99)

With regard to the application of the model for building theory in path research, considerable attention must be given to the definition of path dependence in

[23] This search heuristic refers to the notion of local search. Nevertheless, other heuristics may allow to search more globally, such as drawing analogies based on cognitive maps (Gavetti, et al., 2005).

generalized NK models, as it is substantially different to the original definition (Frenken, 2006). While in NK models path dependence often refers to the path on which an agent moves through the landscape, the original concept of path dependence highlights contingencies and small events in the adoption process. Still, the NK model also has been applied, using the narrow definition of path dependence, for examining the influence of complexity and environmental change on organizational learning (Seidel, 2012). In this model, path dependence is described as follows: the organization starts at a random initial position on the landscape, guided by small events and a self-reinforcing learning process, before finally reaching a suboptimal local peak in the landscape, interpreted as lock-in. In a similar vein, evolutionary accounts examine co-evolutionary path dependence processes over multiple NK-landscapes for extending theory on path formation (Bassanini & Giovanni, 2001).

Compared to system dynamics models, agent based simulation models are in general suitable to capture an emergent phenomena, like path dependence. As the focus of cellular automata is to explain the emergence of macro patterns from spatial micro interactions and how these patterns may change, it is an appropriate method for researching the formation of paths (Davis, et al., 2007). Following this, a formal agent based model based on the cellular automata approach will be derived. The next section will discuss existing models in management research with regard to their ability to capture the research objective.

4.2 Selecting an Appropriate Simulation Approach

In general, social scientists may build agent based simulation models either by starting from scratch or by extending already existing models (Davis, et al., 2007). A simulation-based research program may start with a simple model and then further elaborate it. For example, the basic NK model was first introduced by Kauffman (1993) and then subsequently expanded to explore, for instance, the effects of modularity on coordinating search (Baumann, 2008), the power of analogies in new environments (Gavetti, et al., 2005), or balancing inertia and innovation (Hodgson & Knudsen, 2006). This simulation modeling procedure might be referred to as a building block approach, which amounts to adding complexity in a stepwise fashion to prior computer simulation models (J. R. Harrison, et al., 2007). Instead of beginning with an extensive simulation model, the focus is first put on the most important mechanism and the outcome of the simulation is observed. This enables the researcher to understand the basic behavior of a model and qualifies him to study the consequences of more complex processes, by adding more features, afterwards. In addition, taking a well-known and empirical proven model as reference for simulation work strengthens the external validity of further extensions (Axelrod, 2003). Furthermore, Ethiraj and Levinthal (2009) argue, that adapting a simulation model, instead of building it from the ground up, has the advantage, that the properties of the simulation model are well explored and, that prior simulation modeling efforts facilitates comparison of the results.

A well-known and empirical proven model, incorporating path dependence, is the organizational learning model of March (1991). Prior research extended the March model in order to investigate the effects of bounded rationality on the trade-off between exploration and exploitation (K. D. Miller & Martignoni, 2011), the impact of information technologies on organizational learning (G. C. Kane & Alavi, 2007), organizational processes that affect the variation and retention of knowledge (Rodan, 2005), different philosophical epistemologies in social interactions (K. D. Miller & Lin, 2010), the influence of tacit knowledge on organizational learning (K. D. Miller, et al., 2006), internal variety and environmental dynamism (Kim & Rhee, 2009), or semi-isolation of groups in organizational learning (Fang, et al., 2010). As the organizational learning model already depicts a path dependent process, it may be used as a first starting point for further modeling efforts.

4.2.1 The March Model: Simulating Organizational Learning

By using an agent based computer simulation approach, James G. March (1991) explored the conflicting aspects in organizational learning between the exploitation of current capabilities or competencies and the exploration of new opportunities. While exploration is characterized by terms like path creation, unlocking, experimentation, distant search, organizational slack, and radical innovation, the term exploitation includes path dependence, selection, optimization, execution, operations, or incremental innovation (He & Wong, 2004; March, 1991). Engaging only in one activity, exploration or exploitation, affects firm performance negatively, as they either *"suffer the cost of experimentation without gaining many of its benefits"* (March, 1991: 71) or get *"trapped in suboptimal stable equilibria"* (March, 1991: 71). As firm resources are limited, an organization must strike a balance between the opposing forces of exploration and exploitation (Gupta, et al., 2008). Because returns from exploration are less risky and revenues are realized earlier, organizations and its managers favor exploitation over exploration (Greve, 2007). The returns from this activity then further reinforce the exploitation of current competencies. It is this self-reinforcing nature of exploitation, which makes the process of adapting to an environment potentially self-destructive, eventually leading to path dependence (March, 1991). In order to validate the negative long run effects of exploitation, March developed a simulation model and drew on literature of organizational learning. Organizational learning is thereby defined, *"as a change in the organization's knowledge that occurs as a function of experience"* (Argote, 1999: 31). In organizational learning theory, organizations are regarded as adaptive systems, which are shaped by their members through individual learning processes. These individuals are supposed to learn through an experienced based trial and error process (Levinthal & March, 1993). Then again, organizational rules, norms, structure, and standard operating procedures affect learning and the acquisition of knowledge on an individual or group level (Crossan, et al., 1999). The two levels, the organizational as well as the individual, are intertwined, may each hold learning barriers, and influence learning rates (J. Schilling & Kluge, 2009). Taking these considerations into account, the formal model depicts an organization as a set of learning agents, holding beliefs about an environment, and proceeds in five steps:

(1) *The organization is operating in an exogenously given environment.* As a representation for the environment an m-dimensional vector is initially stuffed with randomly assigned values of -1 and 1, taken from an equal distribution. While a value of 1 can be conceived as an environmental condition being present, a value of -1 describes the absence of the same condition. For illustrative purposes, imagine that an environment consists of four distinctive dimensions: consumer power, price sensitivity, competition, and transport infrastructure. A firm may operate in an environment, where consumers have high power (1), are sensitive to increase in prices (1), competition on the market is low (-1), and the transport infrastructure is fully developed (1). It is determined, that the four-dimensional environment vector <11(-1)1> describes the aforementioned environment. If an organization would mirror this vector, it is fully adapted to the environment. In the simulation model, the environment is initialized once at the beginning of each run and remains static in the basic model.

(2) *The organization consists of n agents holding beliefs about the environment.* Instead of modeling the organization as a whole, it is assumed that n agents, indirectly interacting with each other, make up an organization. Just like the environment, agents are delineated as m-dimensional vectors. At the beginning of a simulation run, each agent vector is randomly initialized by assigning a value of -1, 0, or 1, taken from an equal distribution, to each of the m dimensions. Through these values, agents are assumed to possess a set of beliefs about the environment. While 1 and -1 reflect the agent's opinion about the state of an environmental dimension, a value of 0 reflects, that an agent has no opinion or has no knowledge about an environmental dimension. An agent with the associated vector <100(-1)> correctly believes, that consumers have power (1), but wrongly assumes, that the transportation infrastructure is undeveloped (-1). With regard to the price sensitivity (0) or competition (0), the agent has no opinion, respectively does not know about the presence of these two dimensions. As initially all values are assigned randomly, the probability, that a belief of an agent corresponds to the true value of the environment, is on average one third. By learning from other

agents in the organization, the match between an agent's belief set and the environment increases over time.

(3) *Learning in the organization takes place indirectly through an m-dimensional organizational code initialized with zeros on all dimensions*. In every round, agents update in a random order each of their beliefs from an "organizational code" with a probability p_1. The organizational code can be thought of as shared beliefs, culture, or norms in an organization, with the probability p_1 reflecting the influence of the culture on individual agents within the organization. If a dimension of the code contains a zero, agents do not learn from the code on this dimension. As the code is initialized with zeros, agents will not learn from the code at the beginning.

(4) *The organizational code learns from the dominant beliefs of superior individuals in the organization*. To allow for interactions between agents through an organizational code, the code must be updated. In the simulation model, the code is updated through indirectly learning from a group of "superior agents". Superior agents are characterized by having a higher knowledge level, in comparison to the organizational code and other agents in the organization. This means, that superior agents match the environment on more dimensions as the code or other individuals. Specifically, individual or code knowledge is computed by taking the dot product of the agent or code vector with the environment vector and dividing it by the number of dimensions m. Among the group of superior agents, a dominant belief vector is computed by summing up the belief values on each dimension and over all agents. If the sum is positive, the value of the current dimension is set to 1, if the sum is negative, set to -1, if the sum is 0, the value is randomly set to 1 or -1 with equal probability. The dominant belief vector is therefore reflecting the majority view of the superior group in an organization. In each round, the code learns from the dominant belief vector with a probability of $1-(1-\mathrm{p}_2)^{\mathrm{k}}$, where p_2 is the learning rate of the organizational code from the dominant belief vector and k represents the number of individuals holding a dominant belief minus the number of individuals holding a minority belief. As the organizational code only learns from agents who match the environment on more

dimensions, the code knowledge increases over the duration of a simulation run, while the size of the superior group decreases. By that, the code adapts to the exogenous given environment over time. At some point in time, the code stops learning and is reproduced, as there are no more agents with higher knowledge levels in the organization. Because agents update their beliefs by learning from the code, they will eventually share the same beliefs in the organization, so that a stable equilibrium is achieved that cannot be escaped endogenously.

(5) *As measures for the performance of an organization, the average equilibrium knowledge and the organizational code knowledge are computed*. For measuring the outcome of simulation runs and comparing the results for different learning probabilities, two performance measures based on knowledge levels are introduced: average individual knowledge and code knowledge. The knowledge of an individual agent is calculated by taking the dot product of its belief vector with the environment vector and dividing it by the number of dimensions m. Following this, the average individual knowledge is computed by summing up the knowledge of all agents and dividing the sum by the number of agents n. A special case of the average individual knowledge is the average equilibrium knowledge, which is the measured average individual knowledge, when an equilibrium state is achieved. More similar to the individual knowledge, the code knowledge is computed by taking the dot product of the code vector with the environment vector and dividing it by the number of dimensions m. The parameters used in the organizational learning model of March are summarized in Table 5.

Table 5: Simulation parameters used in the March model (March, 1991)

Parameter	Value	Remarks
n	50	Number of agents within an organization
m	30	Number of environmental, agent, and code dimensions
p_1	[0.1, 0.9]	Probability of agents learning from the code
P_2	[0.1, 0.9]	Probability of the code learning from a superior group
it	80	Number of simulation iterations

Based on this model, March conducted virtual experiments, by altering the learning probability parameters p_1 and p_2 within a range of [0.1, 0.9] and observing the effects on average equilibrium knowledge and code knowledge.[24] The findings were, that the highest average equilibrium knowledge is achieved for organizations where agents learn slowly, meaning that the learning probability p_1 is low, from the organizational code, but the code rapidly adapts towards the dominant belief vector, meaning that p_2 is high. While rapid learning on individual level is considered as good, the effects on an organizational level can be detrimental. The results confirm this statement, as organizations with high individual learning rates achieve equilibrium faster, but eventually, attain lower knowledge levels as compared to slow learners (Figure 11). This effect is caused by the suppression of superior beliefs through the organizational code, as the higher the learning rate of agents, the more likely it is, that a superior agent adopts beliefs inferior to its current set of beliefs.

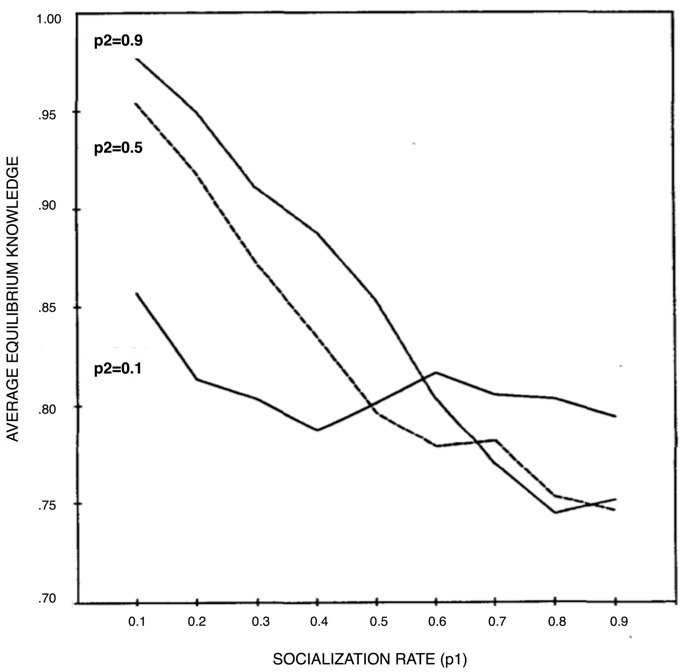


Figure 11: Achieved average equilibrium knowledge levels for different socialization and code learning rates (March, 1991: 76)

[24] These values represent slow (0.1) and fast (0.9) learning. A value of 0.0 would lead to no learning at all, while a value of 1.0 leads to an immediate update of the belief vector.

The results call attention to a fundamental trade-off that organizations face. Either, they rapidly adapt towards an environment, potentially exploiting economics of scale, network effects, first mover advantages, and early learning curve effects, or, they spend more time exploring the possibilities and opportunities within their environment, eventually ending up with a higher performance, but achieving equilibrium slower (Suarez & Lanzolla, 2007). Because these options might be mutually exclusive, an organization has to strike a balance between these counteracting processes over time for being successful in the long run (Gupta, et al., 2008). But, as explorative activities like fundamental research, product development, or corporate venturing are linked to uncertain results, and returns are often lying in distant future, organizations tend to favor exploitation over exploration (Benner & Tushman, 2002; Uotila, et al., 2009).[25] This is, because in contrast to exploration, exploitation generates immediate returns. At the same time, exploitation exhibits increasing returns, therefore potentially leading to path dependence (March, 1991). Indeed, March's simple simulation model includes properties of path dependence and is suitable as a simple representation for a path formation process. At the beginning the outcome of the process is not known, and the scope of action is broad, as heterogeneous agents initially exhibit variety within the organization. As agents begin to learn from the code and get assimilated towards the organizational code, the average individual knowledge increases. But, at the same time, the scope of choice narrows down, as the internal variety of beliefs decreases (Kim & Rhee, 2009). In the end, agents converge to a homogenous set of beliefs and are locked in to reproduce the stable equilibrium.

Compatible with the building block approach, March extended the base model successively and included personnel turnover, incremental environmental change, and heterogeneous learning rates. One notable result of the extensions is, that in a changing environment only personnel turnover counteracts the degeneration of organizational knowledge and allows for escaping the stable equilibrium (Figure

[25] Scholars also highlight the negative effects of putting too much weight on exploration (Nohria & Gulati, 1996). Search processes may be aborted to early, attributed as unsuccessful, and, because of that, further search takes place (Levinthal & March, 1993). The problem here could be more described as path-independence, because over-exploration is increasing the scope of choices.

12). Turnover is constituted in the model by randomly replacing an agent in the organization with a turnover probability p_3. A reason for knowledge detoriation is, that over time agents converge to one single view of the environment. In the case of change, and without turnover, agents still hold on to this homogenous view, impeding necessary adaptation processes. Inducing agents with random beliefs from outside into the organization, increases the variety and enables the organization to adapt. Yet, according to the law of requisite variety, the degree of adaptation depends on the amount of imported new beliefs (Ashby, 1956).

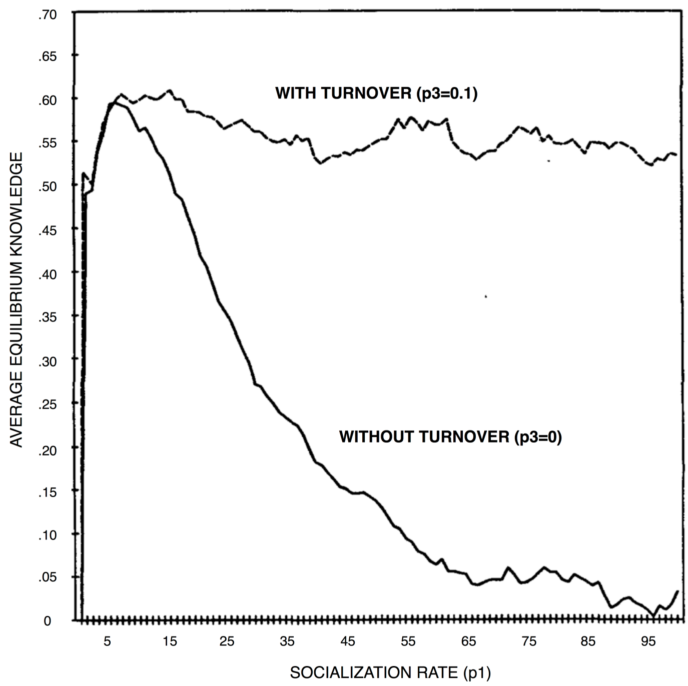


Figure 12: Achieved average code knowledge for learning rates of p_1=p_2=0.5 in the presence of environmental change with and without turnover (March, 1991: 80)

Transferring this proposition to the concept of path dependence would lead to the conclusion, that organizational paths can only be escaped by inducing variety through agents with heterogeneous beliefs from outside of the organization. But, as case studies have shown, paths can also be unlocked endogenously (Burgelman, 1991). Therefore, this assumption has to be neglected. Although the simulation model excludes a proper representation for unlocking, it was an early attempt to capture the logic of organizational path dependence through a self-reinforcing learning process.[26] The simplicity of the model notwithstanding, it was extensively expanded over the last decades. While the original model possesses the aesthetics of a simple model, it assumed an over-simplistic view on an individual learning processes and organizational structure. Recent extensions of the March model suspended the restrictions in the learning mechanisms and used a more realistic representation of an organization by including a spatial dimension.

4.2.2 Extending March: A Model with Interpersonal Learning

In March's model, individual learning takes place indirectly through an organizational code and does not account for the spatial dimension of learning. Yet, the location or prior knowledge of an organization or individual is important for effective learning (W. D. Cohen & Levinthal, 1990; Levinthal & March, 1993). Instead, March's simulation model reflects a hierarchical learning scheme and does not allow for emergent properties, evoked by direct interactions between agents. In a strict sense, the global nature of the code dictates agents in the organization, while local interactions and organizational niches are neglected. But, these local interactions influence the learning process and are primary drivers for the emergence of path dependence (Anderlini & Ianni, 1996). In a similar vein, Garud and Karnoe (2003) argue, that the regional learning process in the Danish wind turbine industry lead to the emergence of a new technological path, and to the unlocking of an old path, through what they call "distributed entrepreneurship". With regard to Garud and Karnoe (2003), a model of path dependence and unlocking must account for location and the spatial dimension of learning. In order to include location in the organizational learning mode of March, it has to be extended. Recent studies suspended the limitation of the March model by including direct learning through an interpersonal learning mechanism (see K. D. Miller, et al. (2006), K. D. Miller and Lin (2010), Fang, et al. (2010) and Kim and Rhee (2009)). In contrast to the learning model with organizational code, direct

[26] March explicitly refers to David (1990) and Arthur (1984) with regard to the detrimental effects of increasing returns and local feedback processes in learning by experience.

learning is integrated in these models, by linking agents with each other either through a structure like a small-world network (Fang, et al., 2010), or by placing them in a cellular automata grid (Kim & Rhee, 2009; K. D. Miller & Lin, 2010). As outlined by Hegselmann (1996), the cellular automata grid is well suited to observe emergent effects of local interactions in social systems. When using a cellular automata approach, agents are positioned in a square grid, consisting of n cells. Each agent occupies one of the n cells. Compatible with the von Neumann neighborhood of the cellular automata, agents may update their beliefs from one of their four direct neighbors.[27] This is in line with the behavioral concepts of myopia in learning (Levinthal & March, 1993) and local search (Levinthal, 1997). Learning from distant agents, by means of local search, needs the dissemination of beliefs to direct neighbors. The restriction to local neighbors slows down the learning process, in comparison to learning from an organizational code, where knowledge can be immediately distributed within the organization. The structure of an extended organizational learning model, taking into account the spatial dimension, is now briefly explained. Like in the March model, it is assumed that an organization is operating in an m-dimensional environment. Furthermore, an organization consists of n agents, each of them holding a set of beliefs about the environment. Despite these similarities, the interpersonal model has some unique characteristics. Subsequently, the procedure of the formal model is explained in three consecutive steps:

(1) *The environment and the organization consisting of n agents are created.* A square grid with edge length $\sqrt{n}$ is constructed, spanning a total of n cells. Each of the n cells accommodates one agent vector. The agent vectors are initially stuffed with random values of -1, 0, and 1 picked from an equal distribution. As in the March model, these values represent beliefs about the properties of an exogenous given m-dimensional environment. The grid is borderless, meaning that every cell, and therefore every agent, shares the same amount of neighbors in the grid. Neighbors are here defined as agents bordering north, east, south, and west of the

[27] Direct neighbors are those connected to the agent in the south, north, west and east.

agent in the grid (von Neumann neighborhood with a Manhattan distance of r=1). Agents that are positioned at the edge of the grid are hence able to learn from at least one agent from the opposite edge side.

(2) *Agents update their beliefs by learning from their direct neighbors.* Each round, every agent learns from the best performing neighbor with a probability of p_1. The learning sequence is kept random, preventing the emergence of misleading patterns. The knowledge level of an agent is used as a performance measurement and evaluated through computing the degree of accordance between an agent and the environment. In the case where more than one neighbor shares the highest knowledge value, an agent randomly selects one of these neighbors for updating its belief vector. In addition, updating only takes place if the knowledge value of the neighbor is higher than the knowledge of the selected agent. If the knowledge is lower, the agent refrains from updating its belief vector in the current round.

(3) *Measurement of organizational performance.* As there is no organizational code, the performance of the organization is evaluated using the average individual knowledge measurement. The average individual knowledge is calculated by summing up all individual knowledge values, divided through the number of agents in the organization. The individual knowledge is, as in the March model, computed by taking the dot product of the agent and the environment vector and dividing it by the number of dimensions. The parameters used in the simulation model are summarized in Table 6.

Table 6: Simulation parameters in the interpersonal organizational learning model used by Miller et al. (2006)

Parameter	Value	Remarks
n	100	Number of agents within an organization
m	150	Number of environment and agent dimensions
p_4	[0, 1]	Probability of agents learning from one of its neighbors
it	100	Number of simulation iterations

Based on the interpersonal model derived above, virtual experiments are conducted by K. D. Miller et al. (2006). In Miller's experiments the learning probability parameter p_4 is set to [0.1, 0.3, 0.5, 0.7, 0.9] and the average individual knowledge is measured. In general, the findings of the interpersonal learning model are confirming the exploration versus exploitation trade-off hypothesis (March, 1991; K. D. Miller & Martignoni, 2011; K. D. Miller, et al., 2006). Higher individual learning rates (p_4) achieve equilibrium faster, but compared to slow learning rates, attain lower knowledge levels (see Figure 13). Yet, the impact of slow [0.1, 0.3] and fast [0.7, 0.9] learning rates is not as pronounced as for the original model.

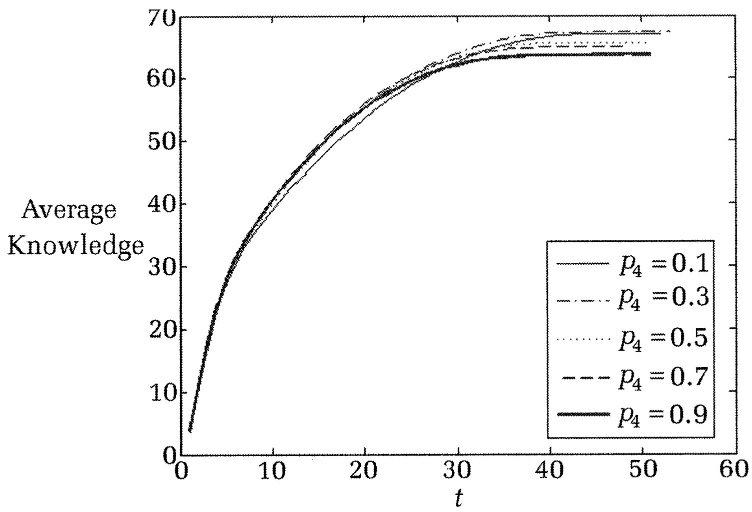


Figure 13: Average individual knowledge for learning rates ranging from slow to fast learning in the interpersonal learning model (Miller et. al., 2006: 715)

Compared to the March model, the interpersonal learning model allows deeper drilling into the connection between spatial structure, path dependence, and organizational learning.[28] But again, while prior interpersonal learning models reflect how an organization adapts through an individual learning process towards an exogenous environment, it does not include the notion of unlocking. Initial

[28] See K.D. Miller & Martignoni (2011) for a review on interpersonal learning models.

efforts to examine the role of variety (Kim & Rhee, 2009) and diversity (K. D. Miller & Martignoni, 2011) were made, but did not refer to the unlocking of organizational paths. Nevertheless, as the interpersonal model allows illustrating the path formation process and can include concepts of diversity, it serves well as a starting point for the integration of unlocking into the model and evaluation of different means to unlock paths.

5. Development of a Formal Model

Common sense suggests, that individuals are not similar in every facet and may disagree with each other. Therefore, it cannot be assumed that individuals are assimilated towards one unique view of the world (Jackson, et al., 1991). Even so, most of the interpersonal simulation models assume, that in a stable equilibrium only one belief set is reproduced. The occurrence of multiple stable equilibriums, at the same time, is hence excluded. Furthermore, as already emphasized in the organizational model of path dependence, it can be doubtful that locked-in organizations consist of organizational members with homogenous beliefs (Sydow, et al., 2009). For instance, Tripsas & Gavetti (2000) argue, that within Polaroid multiple views, and therefore potentially multiple equilibriums, on the future of digital imaging were present. But in the end, adaptation failed, because the top management was holding wrong cognitive representations about the environment and suppressed opinions deviating from the, in their view dominant, razor-blade business model. As most simulation models don't take into account such diversity in beliefs during the lock-in phase, they also do not account for endogenous unlocking of organizational paths. Therefore, important drivers of unlocking path dependence, such as cognitive dissonance, individual heterogeneity, and variety in practicing routines, are missing in current models (Castaldi & Dosi, 2005). For example, with Intel it was the culture allowing for dissonance that eventually unlocked the "memory path" (Burgelman & Grove, 1996), and with Polaroid it was the suppression of divergent views with regard to the business model, inhibiting the unlocking of paths (Tripsas & Gavetti, 2000). A model which wants to examine the phenomenon of unlocking has to integrate heterogeneity in the beliefs of agents or among different groups of an organization, even in the presence of lock-in. Furthermore, it has to include, that individuals are inherently different and therefore, organizations naturally exhibit internal variety (Fiol, 1994; Kim & Rhee, 2009). Dosi & Winter (2000: 5) put it more vividly, by stating that *"...a model without heterogeneity is like a flower garden without color"*. Therefore, in order to incorporate heterogeneity, the underlying logic of agents has to be more sophisticated, than purely relying on the assimilation of each organizational member towards a unique view. While the March model already includes the turnover as a mean to induce variety in the organization, which restores the adaptive capability of an organization in presence of change, it still misses some important means to unlock paths. These means include, for instance, the reconfiguration of the organizational structure (Biggart, 1977; Karim & Mitchell, 2000), influence of a top management team (Beckmann & Burton, 2008), or changes in the corporate strategy (Burgelman & Grove, 1996). Additionally, the models assume, that agents are seeking to maximize their performance with regard to an exogenous given environment by learning from superior agents within the organization. Furthermore, while the speed of learning has an impact on firm performance, the models neglect to incorporate if agents are even able to learn from other, potentially superior, agents. Learning may be difficult or hardly possible if cognitive distance between the beliefs of agents is too large, eventually impeding to tap into superior beliefs of other individuals or groups (W. D. Cohen & Levinthal, 1990; Nooteboom, 2007). Also, the physical distance between agents in the organization could hamper efficient interactions for exchanging beliefs (Nonaka & von Krogh, 2009). It simply may not be possible for individuals in corporations to learn from far off agents in a different department, at least not face to face. A more realistic understanding of how individuals learn and update their beliefs has to include such barriers, by constraining the potential sources of learning to the proximity of the agents set of beliefs (Malmberg & Maskell, 2006; Wong, 2004). Lastly, it is an overly simplistic representation of an organization, when not accounting for different business units, groups, or cultures within organizations. Including that corporations consist of different groups, could impede learning processes (Schein, 2010), or result in recombination of knowledge (Fang, et al., 2010). It is critical, then, to extend the basic interpersonal model with regard to the behavior of agents in the model and means to unlock paths.

5.1 Adapting the Behavior of Agents

In an organizational learning model individuals are assumed to update their beliefs by a process of partner selection and subsequent learning from the selected partner (Dodgson, 1993). While in the basic interpersonal model selection takes place based on maximizing the individual knowledge criteria and learning rates are fixed, the model at hand will include a selection and learning rule, based on knowledge and the similarity of agents. In empirical studies on individual and organizational learning, it has been shown that similarity is an important factor for knowledge sharing (Darr & Kurtzberg, 2000). Such an updating process might explain how heterogeneity emerges on organizational level. Both parts of the process (*selection* and *updating*) will be discussed to derive a simple, but realistic, algorithm for the simulation model.

5.1.1 Selection of a Learning Partner

Economic rational-agent models highlight the utility maximizing behavior of actors in organizations or markets (Janssen, 1993). But, as decision makers lack perfect knowledge, because of their bounded rationality, they may not be able to maximize their individual benefit (Camerer, 1998; Simon, 1972). Instead, agents search for new solutions among a set of given alternatives in their direct environment and stop, when a satisficing solution has been found (Gavetti, et al., 2012). In the presence of substantial uncertainty, agents furthermore rely on standard operating rules, narrowing the decision space to neighboring alternatives (Cyert & March, 1963). These ideas on how individuals search in organizations are captured by the behavioral theory of the firm. Inspired by this theory, behavioral models, like the organizational learning model, emerged. Still, in these models, selection of learning partners takes place according to an expected individual or organizational benefit. Organizational learning models take, for instance, the individual knowledge as the decision criteria for whom to learn from. In such models, individuals select the best or a better performing agent and update their beliefs according to the belief set of these superior organizational actors (Kim & Rhee, 2009; Lazer & Friedman, 2007; March, 1991; K. D. Miller & Martignoni, 2011). Reasons to assume that individuals update their beliefs according to an utility maximizing rule are stated by Kenneth and Fahrbach (1999), and include: actors wanting to achieve their ends through obtaining superior beliefs (D. Katz & Kahn, 1978), reducing uncertainty in decision making (Radner, 1986), conserving or attaining their power within the organization (Burns & Stalker, 1961; Pfeffer & Salancik, 1978), or acquiring new knowledge out of pure curiosity (Freedman, 1965).[29] But, Kenneth and Fahrbach (1999) also argue, that individuals are not only maximizing their utility by learning from superior individuals, but furthermore either prefer learning from similar individuals or are unable to learn from distant knowledge bases (see also Byrne (1971), Friedkin and Marsden (1994), and McPherson, et al. (2001)). For instance, in social psychology, similarity is closely connected to attraction and exhibiting a positive relationship (Pfeffer, 1983; B. Schneider, et al., 1995). Experiments showed, that if individuals share the same beliefs, experiences, or opinions with others, they tend to like them more, are more willing to exchange knowledge with them, and feel personnel excitement (Byrne & Nelson, 1964; Darr & Kurtzberg, 2000). According to O'Reilly (1983), information exchanged with similar others is also more trusted and W. D. Cohen and Levinthal (1990) argue, that successful knowledge transfer occurs with a higher likelihood between similar actors. Additionally, individuals actively seek for information endorsing their decisions and avoid contradicting information sources, increasing the probability for choosing similar individuals to learn from (Festinger, 1950, 1957, 1964). As individuals have difficulties to change existing elements of their beliefs, it proves to be hard to accommodate dissonant information (Starbuck, 1996). Hence, updating from a partner with a similar knowledge base is preferred. For example, Podolny (1994) argues, that similarity is a selection heuristic for individuals operating in uncertain markets. Another reason for learning from similar actors can be found in literature on barriers of individual knowledge exchange (Cabrera & Cabrera, 2002). According to Darr (2000), some common understanding has to be present as a basis for transferring knowledge between actors. Naturally, if an

[29] See Kenneth & Fahrbach (1999) for a detailed overview on why agents adopt superior beliefs.

individual is unable to evaluate new knowledge and recognize its value, because of a large distance in the knowledge base, selection and learning is unlikely (W. D. Cohen & Levinthal, 1990; Schulze & Brojerdi, 2012). As individuals "*perceive, interpret and evaluate the world according to mental categories which they have developed in interaction with their physical and their social/institutional environment*" (Nooteboom, 2000: 71), new information deviating from their current beliefs may not draw their attention. In conclusion, it is reasonable to assume that actors in organizations do not select a peer based solely on performance criteria, but also take into account the similarity of its counterpart. Social systems in which actors update their beliefs in accordance with performance and similarity criteria are referred to as balance and information systems (see Kenneth and Fahrbach (1999) for a definition). Therefore, a selection rule incorporating both criteria includes a factor for performance and similarity. The starting point to derive a similarity-performance selection rule is the computation of individual knowledge (*agent_know*), used in the organizational learning models, where knowledge is simply computed by the dot product of the agent and the reality vector:

$$agent\_know_{n_j} = \begin{pmatrix} b_{n_j,1} \\ \vdots \\ b_{n_j,m} \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix} .$$

Now, instead of taking only *agent_know* as basis for selection, a further factor incorporating similarity is added. Just like calculating the knowledge value, similarity between agents is computed by the dot product of the belief vectors for two neighboring agents n_i and n_j:

$$similarity_{n_i n_j} = \begin{pmatrix} b_{n_i,1} \\ \vdots \\ b_{n_i,m} \end{pmatrix} \cdot \begin{pmatrix} b_{n_j,1} \\ \vdots \\ b_{n_j,m} \end{pmatrix} .$$

The larger the dot product of both vectors, the more the beliefs of the two agents correspond. Integration of balancing between similarity and knowledge is achieved by putting weights adding up to one in front of the statements. Here the parameter *sim* defines, how much weight is put on the similarity aspect and *(1-sim)* defines, how much weight is put on the knowledge aspect*:*

$$balance_{n_i n_j} = (1 - sim)\left\{agent_{know_{n_j}}\right\} + sim\left\{similarity_{n_i n_j}\right\}.$$

An agent now selects the counterpart with the highest *balance* value in its direct neighborhood to learn from. Assuming the grid representation of the interpersonal learning model, a random agent A_i always has four neighbors compared to its own position: one in the north (A_N), one in the east (A_E), one in the south (A_S), and one in the west (A_W). At first, the agent A_i evaluates the *balance* value for each of its four neighbors:

$$balance_{A_i A_N} = (1 - sim)\left\{agent\_know_{A_N}\right\} + sim\left\{similarity_{A_i A_N}\right\}$$

$$balance_{A_i A_E} = (1 - sim)\left\{agent\_know_{A_E}\right\} + sim\left\{similarity_{A_i A_E}\right\}$$

$$balance_{A_i A_S} = (1 - sim)\left\{agent\_know_{A_S}\right\} + sim\left\{similarity_{A_i A_S}\right\}$$

$$balance_{A_i A_W} = (1 - sim)\left\{agent_{know_{A_W}}\right\} + sim\left\{similarity_{A_i A_W}\right\}.$$

Based on the calculated *balance* values, a learning partner is chosen according to the following selection rule:

$$selection_{A_i} = rnd \begin{cases} A_N, & agent\_know_{A_i} < balance_{A_i A_N} \\ A_E, & agent\_know_{A_i} < balance_{A_i A_E} \\ A_S, & agent\_know_{A_i} < balance_{A_i A_S} \\ A_W, & agent\_know_{A_i} < balance_{A_i A_W}. \end{cases}$$

Agent A_i is then randomly selecting a neighbor with a *balance* parameter higher than the agents' individual knowledge. If no superior agent is found in the neighborhood, then A_i suspends the learning process for the current round. Otherwise, the agent updates its belief from the randomly picked neighbor. The updating takes places according to a learning process, through which the organization will adapt to its environment.

5.1.2 Interpersonal Learning within the Organization

After the partner selection procedure has been carried out, learning of beliefs from the selected neighbor takes place. Learning describes the process of knowledge acquisition from individuals and serves on an organizational level as mechanism for adaptation towards an exogenous environment (Gavetti, et al., 2012). The above discussed simulation models from March (1991) and Miller (2006) set learning rates exogenously, to characterize the speed and accuracy with which knowledge is adopted by individuals. Studies applying the organizational learning model, but not focusing on the influence of learning probabilities on emergent outcomes, sometimes keep the learning rates fixed (Fang, et al., 2010; K. D. Miller & Lin, 2010). In this study, the learning probability is not exogenously altered, nor fixed, but endogenously determined by the current state of the model. Research on absorptive capacity implies, that learning performance reaches its maximum when new knowledge is closely related to prior knowledge and decreases depending on the novelty of the knowledge domain (W. D. Cohen & Levinthal, 1990). In line with this concept, learning rates are adjusted with regard to the similarity between actors in the model. The more beliefs are shared, the easier learning takes place between individuals, emphasizing the importance of similarity for the acquaintance of knowledge. Again, the similarity measurement is used to derive the learning probability. The *learning_prob* equation delineates such a simple learning rule:

$$learning\_prob_{n_i n_j} = \frac{similarity_{n_i n_j}}{m}.$$

After the selection and learning process has been formalized, the self-reinforcing mechanisms, necessary to consider a process path dependent, are explained.

5.1.3 Self-Reinforcing Mechanisms

Known self-reinforcing mechanisms leading to path dependence are learning, coordination, complementary, or adaptive expectation effects, where none of the effects must be mutually exclusive (Sydow, et al., 2009). The aforementioned formal model depicts self-reinforcing learning effects through the selection and learning process. While in the original organizational learning models the learning process was governed by one positive feedback loop, namely the self-reinforcing knowledge acquisition, the extended model integrates multiple positive feedback loops (Figure 14). At the beginning of a round, an agent A_i randomly selects another agent with a higher balance value and learns from the chosen agent, with the learning rate depending on the similarity value. Through the updating of the beliefs, A_i assimilates towards the other agent, increasing on the one hand the similarity value, and on the other hand obtaining new knowledge. As knowledge depends on the accordance of individual beliefs with the environment, the external fit of the organization increases as well, resulting in higher firm performance.

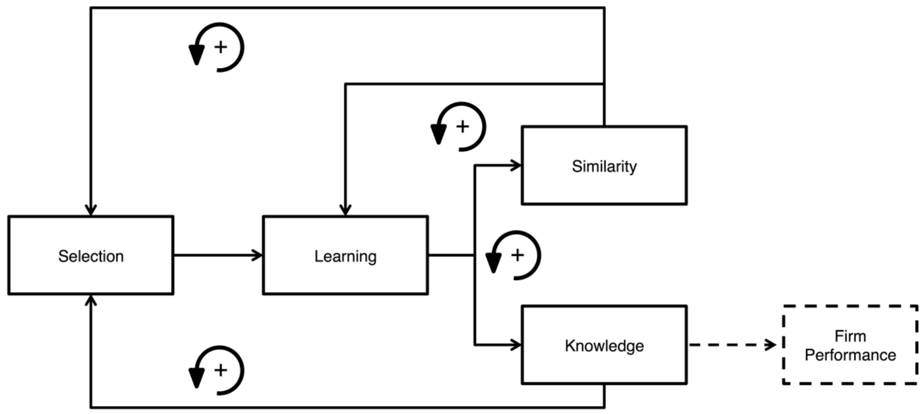


Figure 14: Self-reinforcing mechanisms in the selection and learning process

Beyond that, with an increase in similarity the probability of learning also increases, meaning that agent A_i will on average update more beliefs compared to the previous round. Furthermore, as similarity and individual knowledge increases, the probability of maintaining the selected learning partner, the strength of the bond between the two agents, increases as well. This cycle continues, until the interaction pattern eventually becomes fixed. As now not only knowledge levels are used as selection criteria for whom to learn from, but also similarity, it could be the case, that in the equilibrium state of the model no shared understanding over all agents exists. This may result in heterogeneous views within the organization, potentially facilitating the unlocking of paths. But, as put forth by Vergne and Durand (2010), unlocking paths also needs an exogenous shock. The next section will therefore address, how environmental change, and specifically an environmental shock, is integrated into a formal model.

5.2 Integrating an Exogenous Shock

Organizations face the problem of changing environments (Duncan, 1972). Reasons for turbulence in the environment are, for example, the emergence of new technologies, changes in consumer behavior, or political instability (Kraatz, 1998). While environmental change may provide new business opportunities for an organization, it can also be detrimental for previously acquired competencies

(Tushman & Anderson, 1986). Organizations may find, that their existing competencies, beliefs, or cultures are not suitable for environmental adaptation, and may even become obsolete or “core rigidities” (see for examples Kiesler and Sproull (1982), Leonard-Barton (1992), and Moorman and Miner (1997)). In the absence of an exogenous shock, path dependence may be of no concern for organizations, or even highly efficient (Barney, 1991, 2001). It may be only when the environment changes, that path dependence can become a problem (Sydow, et al., 2009). In accordance with the definition of path dependence and punctuated equilibrium, environmental change in the form of an exogenous shock is a necessary condition to unlock paths (Vergne & Durand, 2010).

5.2.1 Definition of an Exogenous Shock

An exogenous shock contrasts to other forms of environmental change according to four dimensions: frequency, amplitude, speed, and scope (Suarez & Oliva, 2005). Frequency describes how often changes in the environment occur, amplitude describes the magnitude of deviation from the changed environment compared to the initial or previous environment, speed describes the rate of change, and scope describes the number of dimensions, which are affected by simultaneous disturbances. With these four dimensions, five types of environmental change can be distinguished (Table 7).

Table 7: Typology of environmental change according to Suarez & Oliva (2005: 1022)

Frequency	Amplitude	Speed	Scope	Type of environmental change
Low	Low	Low	Low	Regular
High	Low	High	Low	Hyperturbulence
Low	High	High	Low	Specific Shock
Low	High	Low	Low	Disruptive
Low	High	High	High	Avalanche

While the influence of different types of environmental change on a path process may differ, the focus here is put on the specific shock. Using the classification of Suarez & Oliva (2005), a specific shock occurs rarely (low frequency), but abruptly (high speed), with a significant impact (high amplitude), in one specific industry sector (low scope). With this definition, the exogenous shock can be integrated into the model.

5.2.2 Integration of an Exogenous Shock

As the basic organizational learning model only accounts for incremental change, the model must be extended with the possibility to depict an exogenous shock. Furthermore, the constituting dimensions for change need to be transferred into the formal model. As the model only looks at one organization, and not at multiple organizations operating in different industries, the parameter for scope of change is dispensable. The frequency, speed, and amplitude dimensions are captured by the four parameters stated in Table 8.

Table 8: Parameters for extending the simulation model with environmental change

Parameter	Dimension	Value range	Remarks
turbulenceRate	Frequency	[0, 1]	States the probability of change on each of the environmental dimensions.
pointOfChange	Frequency	[1, steps]	Defines specific points during the simulation run at which environmental change is initiated.
turbulenceRange	Speed	[1, steps]	Defines the duration of environmental change in periods.
dimensions	Amplitude	[1, size]	Defines the number of dimensions that may be changed.

Using these parameters, different types of environmental change can be modeled, and the influence on a path process may be examined. For the exogenous shock, the parameters are defined as follows. The probability that change occurs is set to one (*turbulenceRate=1)*. As the shock occurs only once during a simulation run, there is only one *pointOfChange*. An exact value for *pointOfChange* is derived later through parameter variation. As put forth by Suarez and Oliva (2005), an exogenous shock happens immediately and takes place rapidly. In a formal model it is therefore assumed, that a shock takes one period (*turbulenceRange=1*). Furthermore, an exogenous shock affects all dimensions of the environment with equal probability (*dimensions=size)*. Next, with turnover and reconfiguration, two intentional means for unlocking will be derived and included in the formal model. Turnover has been selected to examine how the invasion of new beliefs and heterogeneity affect unlocking, while reconfiguration illustrates the reallocation of resources through a process of rotating agents in the organization (see Table 4).

5.3 Integrating Turnover and Reconfiguration

As previously investigated in the original March model, turnover of agents is a strong option to induce heterogeneity into an organization for facilitating adaptation (Alexander, et al., 1995; March, 1991). While the original model only examined turnover interplaying with incremental change, here the focus is put on the influence of turnover in the situation of an exogenous shock, and with a similarity based selection and learning mechanism.

5.3.1 Turnover of Agents

A considerable amount of organizational knowledge exists in the beliefs of individual members. For this reason, labor turnover can hamper the retention of knowledge in an organization (Hollenbeck, et al., 1995). If knowledge rests solely in individuals, organizational lock-in should be easily overcome by selecting individuals with obsolete knowledge and replacing them with individuals bringing in new beliefs (March, 1991). However, knowledge is not only embedded in individuals, but also in the interactions between them. Knowledge embedded in interactions is less likely to depreciate, compared to knowledge possessed by individuals (see for examples Argote (1999), Argote, et al. (1990), and M. D. Cohen and Bacdayan (1994)). Knowledge embedded in interactions is thus more likely to survive turnover. Even so, labor turnover may provide a reframing of current views in the organization and eventually lead to change (March, 1981). A similar mean as turnover is the reconfiguration of organizational structure through the rotation of agents within the organization.

5.3.2 Reconfiguration

In literature on organizational renewal, reconfiguration, restructuring, or reorganization is seen as a mean to overcome inertia, which is closely related to the concept of path dependence (Bowman & Singh, 2007; Zajac & Kraatz, 2007). For example, research on moving organizational members across subunits suggests, that member rotation can be an instrument for stimulating the formation of new beliefs in groups, provided that they share some common practices (A. A. Kane, et al., 2005). While this may hint to a potential candidate for unlocking a path, one has to bear in mind the differences between the formal and informal structure of an organization. Managers have the ability to set a formal structure in organizations by assigning actors to certain tasks. In particular, the formal structure is defined through job roles, responsibilities, and communication structures between actors in an organization (Child, 1972). Managers may change this composition in their role as organizational architects (Jacobides, 2006). The resulting recombination may facilitate the flow of information and stop dysfunctional routines, allowing for the formation of new beliefs (Jacobides, 2006; Simon, 1962). Influenced by the formal structure, an informal structure emerges through unintended interaction patterns between actors. While managers could rapidly alter the formal organization, the informal organization may not be affected to the same extent (D. Miller & Friesen, 1984; Nickerson & Zenger, 2002). Therefore, interactions between individuals may perpetuate informally, though the formal structure already has been changed. This is in line with the argument by Sydow et al. (2009), that path dependence develops behind the back of actors and is governed by hidden rules. To examine the possibility of unlocking path dependence by restructuring through the moving of members, the focus is here put on the informal interaction processes in the organization.

5.3.3 Integration of Turnover and Reconfiguration

In the original model, turnover is achieved by randomly picking an agent and replacing it with a probability p_3 through a new agent with a randomly generated set of beliefs (March, 1991). In an interpersonal learning model, with a grid representing the organization, turnover is integrated similar: a cell of the grid is randomly selected and the associated agent is replaced with a defined probability. While with turnover a new agent from outside joins the organization, reconfiguration swaps the position of two agents inside the organization. The rotation of organizational members is achieved by randomly selecting an agent within the organization and rotating it with a random other agent in the organization. Table 9 gives a summary of the parameters used for turnover and rotation.

Table 9: Parameters for the simulation of turnover and rotation

Parameter	Value Range	Remarks
turnoverRate	[0, 1]	Probability of an agent being replaced after an exogenous shock occurred
rotateRate	[0, 1]	Percentage of agents changing position within the organization

Up to now, the organization does only contain one hierarchical level. This restriction is suspended and a second hierarchical level, consisting of a top management team, is included in the model. The influence of a hierarchy on path formation is emphasized by Petermann, et al. (2012), but the effects of a hierarchy on unlocking need further elaboration, as changes in the top management team may result in the unlocking of an organizational path.

5.4 Integrating a Top Management Team

The top management team determines the goals, rules, strategies, and structure of an organization (Pfeffer & Salancik, 1978). Because of the top management teams influence, power within an organization is assigned to those, who are most suitable to manage the current problems of the organization (Thompson, 1967). But, in the case of environmental turbulence and decline in firm performance, the power structure in organizations may shift. Power is then assigned to individuals, who are managing the interest of the company in the new context best (J. R. Harrison, 2008). Such changes in the composition of the management team during a crisis might result in an overall corporate strategic change (Wiersema & Bantel, 1992). This is, because in a hierarchic organization the interpretation of the environment is strongly influenced by the management. Hence, bringing in members with new cognitive models about the world, could unlock existing organizational paths (Wiersema & Bantel, 1993). For example, an empirical study on Liz Claiborne, apparel retailer and manufacturer, revealed, that changes in the composition of the management team lead to the unlocking of an organizational path (Siggelkow, 2001). The possibility of unlocking paths through the order of a top management team is taken into account, by including a hierarchical level in the simulation model. More specific, the top management team is integrated into the model, based on prior modeling efforts of top management teams and dominant coalitions (J. R. Harrison, 2008; K. D. Miller & Lin, 2010). Following this literature, the top management team, and its influence on the organization, is integrated in four consecutive steps:

(1) The top management team is funded at the beginning of each simulation run. Through the parameter *bestPercentage,* a defined percentage of agents with the highest individual knowledge levels in the organization are assigned the status of a top management team member. For example, if *bestPercentage* = 0.05 and n=100, the five agents with the highest knowledge value in the organization are assigned the status of a top management team member.

(2) In each round, members of the top management team negotiate a firm strategy

by using a simple majority rule. The rule proceeds as follows: on each dimension beliefs are summed up and divided by the number of top management team members. The number of top management team members is computed by multiplying the parameters *bestPercentage* with n. If the sum on one dimension is greater than zero, a value of 1 is assigned to a strategy vector. When the sum is less than zero, a value of -1 is assigned. Otherwise, 0 is given. Repeating the procedure for each of the m dimensions, implicates an m-dimensional strategy vector.

(3) After learning from their neighbors, agents in the organization update their beliefs from the strategy vector with *learningProbabilityFromTMT*. Two parameters capture the updating probabilities from the strategy vector, before (*learningProbabilityFromTMTbefore*) and after (*learningProbabilityFromTMTafter*) the shock in the environment occurred.

(4) In the case of an exogenous shock in the environment, the top management team may be reconfigured by exchanging current members, either with successors from within the organization or randomly generated agents from outside of the organization. The exchange type of top management team agents is captured by the parameter *takeBestforTMT*. Additionally, the parameter *turnoverTMT* sets the number of agents being exchanged after the exogenous shock. The top management team simulation parameters are listed in Table 10.

Table 10: Parameters for the simulation of the top management team

Parameter	Value Range	Remarks
learningProbabilityFromTMTBefore	[0, 1]	Probability of agents learning from the strategy vector, before an exogenous shock.
learningProbabilityFromTMTAfter	[0, 1]	Probability of agents learning from the strategy vector, after an exogenous shock.
turnoverTMT	[0, bestPercentage*n]	Number of agents getting turned over once, after the exogenous shock occurred.
bestPercentage	[0, 1]	Percentage of best agents assigned to the TMT

takeBestforTMT	[true, false]	*true:* Top management agents get replaced by the best agent within the organization *false:* Top management agents get replaced by random agent from outside of the organization

To evaluate and analyze the influence of similarity, exogenous shocks, agent rotation, turnover of agents, and top management team influence on the organization and its ability to unlock paths, parameters measuring the outcome of the simulation have to be defined.

5.5 Simulation Measures

Simulation models must include measures to evaluate the outcomes for the different parameter alterations and simulation runs (N. Gilbert & Troitzsch, 2005). Furthermore, without using accurate measures, a simulation may not reveal interesting results. Also, measurement types vary from one simulation framework to another. For example, the NK framework assigns to each point in the landscape a performance value and measures the average performance over all iterations, the number of steps until a peak in the NK landscape is reached, or how many times a global peak is achieved (Baumann, 2008). Organizational learning models, like the one from March (1991), measure performance by comparing agents within the organization to an exogenous given environment and take the average of the sum over all individual knowledge values (K. D. Miller & Lin, 2010). To retain comparability between different learning models, a knowledge based measurement parameter should be maintained (Sargent, 2010). In the derived model, the parameter *sumAvgIndKnowledge* is therefore used to measure the average knowledge of an organization.

As extending the grid representation with a similarity based learning algorithm may allow for a stable equilibrium with heterogeneous beliefs in the organization, two additional measures are integrated to capture these differences between individuals. First, the *groups* measure captures the number of different belief vectors within the organization, and is then a measurement for the number of groups within an organization. For example, a *group* value of two suggests, that there exist two different groups, each holding a unique set of beliefs about the environment. Second, a measurement captures directly the diversity of beliefs within the organization. Different to the *group* measure, the *diversity* measure captures the number of different beliefs in the organization. For example, a value of one suggests, that an organization is holding all possible beliefs and may therefore be capable to match any changes happening in the environment. Belief diversity is computed similar to the diversity measurement from Kim and Rhee (2009) by summing up the number of different beliefs (excluding zero) per dimension and dividing it by 2*n. Only when all agents would not have information about any dimension in the environment, will the *diversity* value drop to zero. Likewise, if the *diversity* value equals 0.5, all agents in the organization share the same set of beliefs.

At last, measures for the unlocking of path dependence are defined. According to Sydow, et al. (2009), it first has to be assured, that the organization is locked into a stable equilibrium in order to capture the process of unlocking. A lock-in is reached, if the organization cannot escape an equilibrium state without exogenous pressure. In the formal model, lock-in is therefore achieved, when the average individual knowledge would remain stable infinitely (see also Seidel (2012) and Meyer (2012) for a similar definition of lock-in in simulation models). To check for stability, a simple measurement algorithm compares the present with the previous average individual knowledge value. If the difference between the two values equals zero over a fixed time frame, it is assumed, that the model remains in a stable equilibrium and lock-in has occurred. After the presence of a stable equilibrium is proven, a measurement for unlocking has to be derived. Unfortunately, the current definition of what is meant by unlocking of organizational paths remains quite fuzzy (Sydow, et al., 2009). Therefore, the minimum definition of unlocking is used, saying, that a superior alternative compared to the state after the exogenous shock has to be implanted in the organization. In the model at hand, a superior alternative would mean, that the average individual knowledge level must exceed a threshold value after the exogenous shock occurred. Therefore, the measure *threshold* defines a value that has to be exceeded a certain period of time after the shock occurred. A second measure, named *num_unlocking,* counts the number of simulation runs over all iterations exceeding this threshold value. Defining the *threshold* value is done by means of parameter variation, carried out in the first experiment. The different measures are summarized in Table 11.

Table 11: Measures for simulation outcomes

Measure	Value Range	Remarks
sumAvgIndKnowledge	[0, 1]	Averages the knowledge over all agents in the organization.
groups	[0, 100]	Number of different belief vectors in the organization.
diversity	[0, 1]	Percentage of different beliefs in the organization.
threshold	[0, 1]	Defines a performance threshold value for when a path is considered to be unlocked.
num_unlocking	[0, 300]	Counts the number of iterations where the performance threshold valued was exceeded.

With the formal model at hand the influence of the individual decision rule on breaking organizational paths may be examined. But, in order to perform computer experiments, the model needs to be transferred into machine-readable code.

5.6 Transferring the Formal Model into Computer Code

Implementation of the model "*should achieve three goals: validity, usability, and extendibility*" (Axelrod, 2003: 7). First, concerning usability, it must be assured that simulation results, simulation procedure, and computer code are in detail and comprehensibly outlined for oneself, and other researchers interested in the simulation model. Second, the computer simulation must be extendible to ensure that others are able to rerun the simulation for checking and verifying results, altering simulation parameters, or extending the model with new features. Third, the computer code must be internally valid, meaning that it has to be an accurate representation of the formal model. Furthermore, the possibility that results are caused by an error or bug in the computer code has to be convincingly ruled out. It should be recalled that validity here refers to the transformation of the model into computer code and does not make any propositions about the accuracy of the model, nor how, or if it does reflect reality (N. Gilbert & Troitzsch, 2005).

Bearing these three goals for programming a simulation in mind, a computational framework for implementing the formal model must be found. In general, there are two distinct approaches for transferring the model into an executable computer program. Either an abstract modeling tool is applied or the model is directly implemented by using a native computer programming language.

5.7.1 Modeling Software versus Native Programming Language

Modeling software tools provide a virtual environment for the implementation of simulation models. Examples for software tools include Microsoft Excel, NetLogo, AnyLogic, Repast, Mason, or Matlab (see Nikolai and Madey (2009) for a comparison of software tools for computer simulations). Most of the modeling software comes with function libraries, sample simulation models, editors, and graphical user interfaces. Some of them even require no prior coding experience. For instance, NetLogo replaces a higher programming language through an abstract representation of agents and interactions between these agents. While the simplicity of modeling tools may facilitate the entry into the world of computer simulations, there are also potential drawbacks. For example, the abstraction inherent in such tools may not allow for extensive models, underlying functions may not be well documented, and rigid frameworks might impede the implementation of elaborated simulation models (Macal & North, 2010). Furthermore, some of the aforementioned simulation software packages are expensive and may be out of scope for doctoral students in the social science.[30] At last, replication and extensibility of the simulation model is restricted to a small group of people, who possess and understand the modeling software.

Some of these limitations can be avoided by using a native programming language, like C#, Python or Java. The downside of this approach is that all of the aforementioned programming languages require, at least some, knowledge about object oriented design patterns and programming skills are needed. On the one hand, using a programming language gives flexibility, as there is fewer restrictions compared to modeling tools and functions are well documented. But, on the other hand, thorough consideration must be given if the effort of learning to code and write functions from scratch pays out. Because it is not often known in advance, how the model will look in the end, an accurate approximation of the costs is hard to obtain from the beginning on. Nevertheless, experienced simulation scientists recommend modelers with coding skills to use a higher programming language like Java (Axelrod, 2003). Java has the advantage of effortlessly switching between coding and testing, and there is plenty of documentation on the programming language itself, as well as tutorials. Java also is widely used, making it easy to get assistance if needed. For these reasons, the simulation model will be implemented by using the Java programming language with the integrated development environment Eclipse Kepler and the subversion tool Tortoise (Figure 15).[31] Eclipse and Tortoise are available for free and facilitate researchers with expertise in Java to easily write, check, and extend computer code.[32] To further increase comprehensibility, the computer code is well commented and split into several classes. The measurement output is displayed in the console window and simultaneously written to an unformatted text file. Through a parameter the output

---

[30] E.g. AnyLogic (www.anylogic.de) is priced at 4,800 € for the standard license (status as of August 2013).

[31] A subversion tool manages changes in files and allows recovering historical versions of the computer code.

[32] Java can be downloaded from www.java.com and Eclipse from www.eclipse.org and is available for all mayor operating systems.

can be adapted to display the results for each time step or to display the average results at the end of each simulation run. Text files are optimized for the import into Microsoft Excel for Mac 2011, which is used to prepare the data for analysis and generating charts.

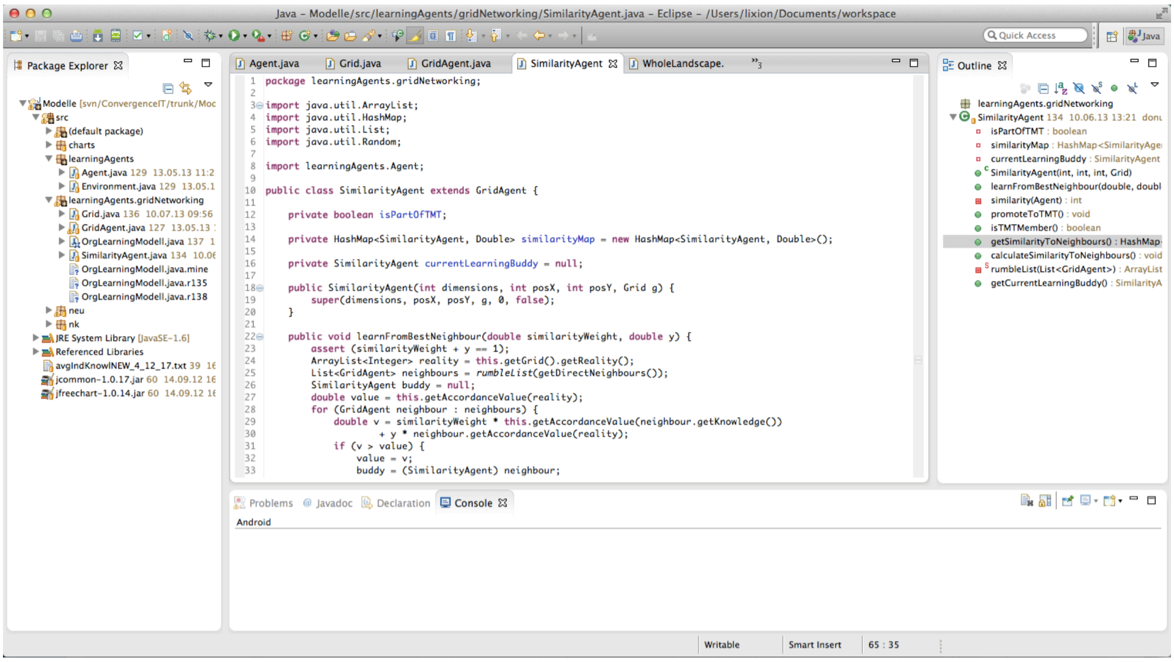


Figure 15: Screenshot of the Java IDE Eclipse

While the choice of a suitable simulation framework can increase extendibility and usability, the issue of internal validity is largely independent from this decision. Therefore, after implementing the formal model into computer code, it is necessary to check for errors and if the implemented simulation is a valid representation of the formal model. Subsequently, a procedure to prohibit and detect errors and ensure the validity is proposed.

5.7.2 Validation of the Simulation Model

Internal validity *“addresses the extent to which a simulation functions in the intended manner”* (Feinstein & Cannon, 2002: 430). Important to note is that internal validity does not make any propositions about how accurate the simulation model captures a real world phenomenon. To guarantee internal validity, it must be assured that the measurement results are due to the behavior of the simulation, and not because of errors in the computer code. But, checking the code for errors is a difficult exercise, as the complexity of simulation models can quickly get overwhelming (Axelrod, 2007). Especially distinguishing if emergent outcomes are based on the behavior of the model or are merely produced by a bug in the code proves to be hard. Hence, there is a great likelihood that the outcomes of the first implemented simulation models are due to computer bugs (N. Gilbert & Troitzsch, 2005). To assure the internal validity of the model implementation, a procedure to identify errors has to be put in place. Left ajar on prior literature of simulation verification and validation (see Axelrod (1997a); Davis, et al. (2007); Feinstein and Cannon (2002); N. Gilbert and Troitzsch (2005); Sargent (2010)), a four-stage framework will ensure the correct implementation of the organizational simulation model:

1. *Replication:* The simulation model is grounded on the organizational learning model of James March (1991). As a starting point, March' original simulation model is replicated exactly and the main findings are compared. If the outcomes of the replication correspond to the original model, the conclusion is drawn that the replicated model is a valid representation of the original model. Only after the basic mechanisms are accurately replicated, the model is extended with interpersonal learning, using the cellular automata approach and sequentially more mechanisms are added.

2. *Robustness checks and calibration:* Robustness checks comprises a family of methods, used to alter the input parameters, while at the same time measuring the changes in the simulation output (Richiardi, et al., 2006). Through this approach, the robustness of the model and its outcomes can be examined and guaranteed. Robustness checks are here conducted in order to ensure that results are stable within a value range. Calibration of the simulation model is used to define the initial parameters. Following the

call from Richiardi, et al. (2006) to use a rigorous calibration method, a systematic approach is applied to set the duration of the simulation, the number of iterations, number of dimensions, and the grid size.

3. *Validation:* Error free code is hard to achieve and seldom possible (Das, 2006). Yet, through means like debugging, testing, and bug fixing, most serious errors may be detected and eliminated. Most of the time, simulation scientists check and correct code to assure that results are not caused by errors (Axelrod, 2003).

4. *Plausibility checks:* Finally, common sense is important to evaluate the validity of the simulation model. Exceptional results should draw the attention of simulation scientists and need to be double-checked in order to avoid wrong conclusions.

As even small errors in the code may have major effects on the simulation outcome, this complex procedure has been chosen. Although bugs may never be entirely ruled out, it is expected that the four-stage approach is a proper mean to detect most of the bugs. Subsequently, the first two stages in the error correcting procedure will be briefly discussed and carried out.

Stage 1: Replication

In order to build upon prior models, simulation models should be replicated to understand the underlying mechanisms. Because of this, the simple organizational learning model of March is replicated before implementing the cellular automata grid model.[33] To do this, the original model is transferred from BASIC into Java code and the resulting average individual knowledge is measured. Figure 16 compares the findings of the March model with the outcomes of the replication.

[33] As the paper of 1991 did not include sufficient information to exactly replicate the results, the original code of the model proved to be helpful. I hereby thank James March for his help by providing the original code in Basic and his remarks on the simulation model.

Even though the model is replicated exactly, small deviations arise due to the low number of iterations used by March and the differences between the two programming languages.[34] Still, all functions are implemented according to the original code, obtained from James March, and the results qualitatively mirror the simulation outcomes.

(a) (b)

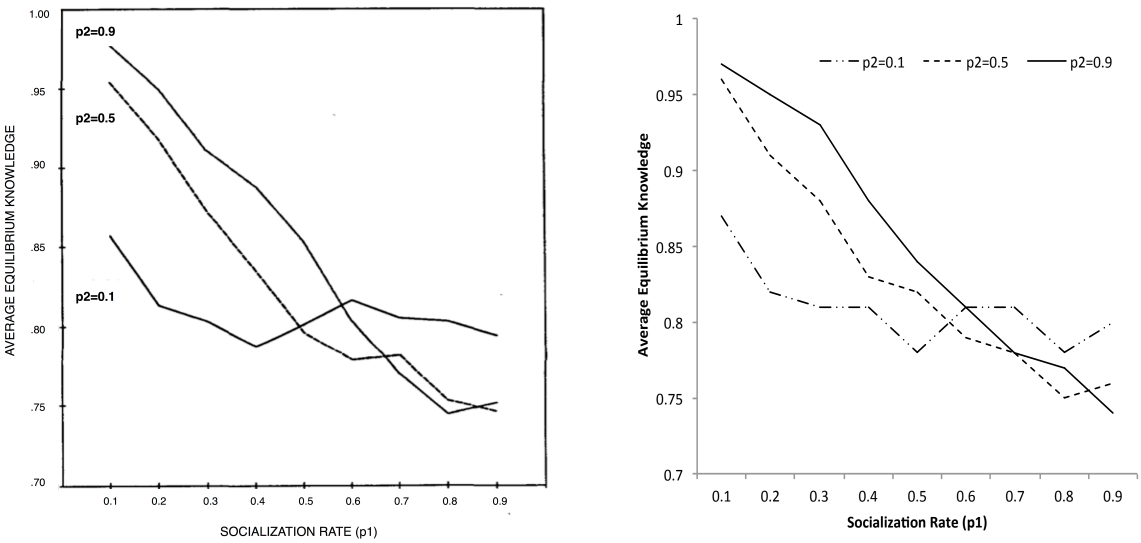


Figure 16: Results of (a) the original March model (March, 1991: 76) and (b) the replication

Therefore, it can be concluded that the replicated model is implemented correctly and can serve as a basis for further extensions.[35] Furthermore, it is assumed that the replicated model does not contain any significant errors. In the next step, the model is extended by assigning each agent a position in a cellular automata grid, instead of learning from an organizational code. For determining the range of the

[34] Random number generators, function calls etc. are just a few examples where programming languages may differ. “Numerical identity” can only be achieved when the same random number generator is used (Axelrod, 1997a). An in-depth comparison of random number generators is out of scope.

[35] Extensions, like personnel turnover or incremental change, are also included in the replication, but graphs are omitted. The results qualitatively mirror the outcome of March’ organizational learning model.

simulation parameters, robustness checks are conducted in the second stage of the protocol.

Stage 2: Robustness Checks and Calibration

By variation of initial settings and simulation parameters the robustness of a simulation model is checked (Richiardi, et al., 2006). When the behavior of the model does not significantly change with variation in initial conditions and parameters, the model is conceived as robust. This increases the credibility of the simulation model (Harrison, 2007). For similarity, rotation, turnover, or top management influence the parameters are altered within a range from [0.01; 1.00] and no extreme outcomes in average individual knowledge were observed. This leads to the conclusion that the model exhibits a robust behavior within this value range. To calibrate the model, the number of iterations, length of the simulation run, size of the belief vector, and size of the grid is determined subsequently.

Determining the Number of Iterations

The required number of repeated simulation runs can be approximated using the dimensionless variation coefficient c_v, defined as the ratio of the standard deviation s to the arithmetic mean μ (Lorscheid, et al., 2011). Beginning with a low number of iterations, the variation coefficient c_v is computed and the number of iterations is increased until it reaches a stable equilibrium. From this point onwards, a further increase in iterations does not contribute to the accuracy of the model, but solely adds computation time. Figure 17 shows c_v for up to 10,000 iterations and the time needed in seconds to conduct the iterations of the simulation. For 50, 100, and 150 iterations, c_v is fluctuating and stability is not achieved. From 300 iterations onwards, c_v can be considered as stable, although there is some deviation when compared to 10,000 iterations.

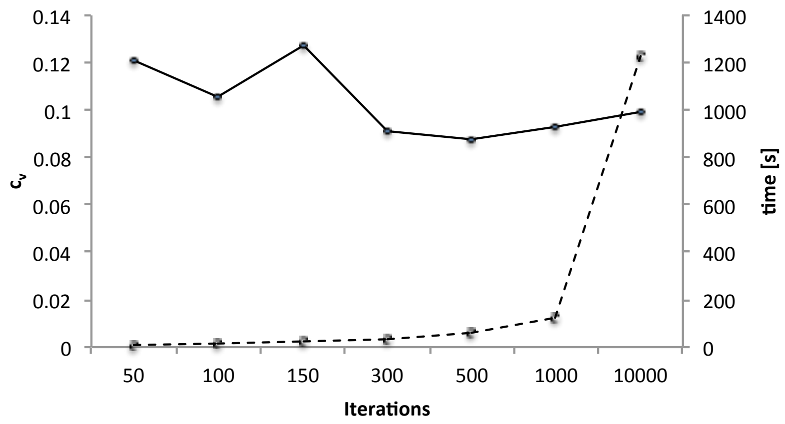


Figure 17: Graphical approximation of iterations according to Lorscheid et al. (2011)

Yet, because of the trade-off between the time it takes to run the simulation and the accuracy of the outcomes, for further examinations 300 iterations have been chosen. Furthermore, as similar models repeat the simulation runs only 50 times (see K. D. Miller and Lin (2010)), the 300 iterations chosen here seem to be fairly sufficient.

Determining the Duration of a Simulation Run

The duration or length of a single simulation run in time steps or ticks must be chosen to allow for a stable equilibrium. Equilibrium is here defined as a stable state, where the system is in balance between the forces inherent in the model. The stable equilibrium at the end of a simulation run also equates to an organizational lock-in of a path dependent process. As measure for the equilibrium the stability of the average individual knowledge is used. It is assumed, that if the knowledge remains unchanged, the underlying interaction pattern between agents is unable to unlock a path. Figure 18 gives the maximum time steps after which the average individual knowledge does not change for similarity values of 0, 0.05, 0.1, 0.15, 0.2, and 0.25 over 300 iterations, with a grid size of 10 and agent size of 75.

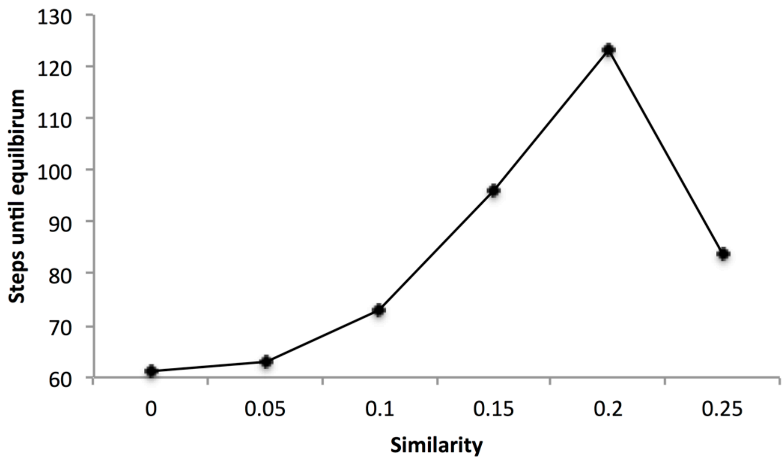


Figure 18: Maximum number of time steps needed until equilibrium is achieved

The maximum number of time steps is 123. Increasing the number of time steps beyond 123 steps will add computation time, but, as the system remains unchanged, not provide further insights. Therefore, the optimal length of the simulation could be set to 123 steps. Yet, as only 300 iterations are conducted, choosing 150 time steps adds a safety factor of approximately 20 percent. As later experiments include an exogenous shock, the total length of the simulation is therefore doubled and set to 300 time steps. This procedure potentially allows achieving a new stable equilibrium after the shock occurred.

Determining Grid Size and Number of Beliefs

In this section, the model is calibrated with regard to the size of the grid and the number of beliefs. More particular, the influence of both parameters on the average individual knowledge and time to equilibrium is investigated. Figure 19 depicts the influence of grid size on the two outcome parameters for grid size values of 2, 5, 10, 15, 20, and 50. The similarity value is kept fixed at zero. The results show that increasing the grid size has a positive effect on average individual knowledge and a negative effect on steps until equilibrium is achieved.

More agents in an organization hence increases firm performance, but slows down the propagation of beliefs.

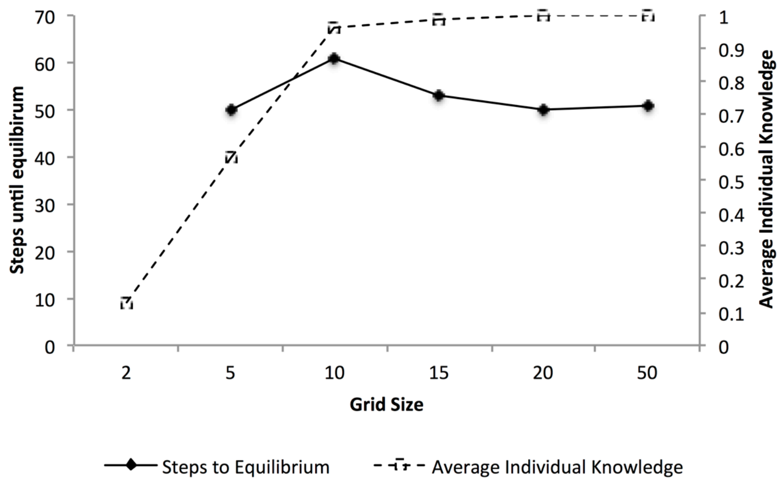


Figure 19: Calibration for the size of the grid holding belief size (m=75) fixed

The finding that information in smaller organizations may be exchanged faster and more directly, as well as that organizations with more agents and possibly higher variety may be able to reach a higher average individual knowledge, sounds plausible. A special characteristic of the simulation model is observed for grid sizes smaller than five. For instance, in some simulation runs, for a grid size of two, no equilibrium is achieved. This is, because agents may be too dissimilar to effectively learn from each other. For grid sizes smaller than ten, the achieved average individual knowledge cannot attain very high values, because the belief variety within the organization is not sufficient for effective adaptation towards the environment. Arguably, enlarging the grid size has on the other hand the drawback of exponentially adding computation time. For example, having a grid size of ten takes approximately one minute for 300 iterations. While setting the grid size to fifty, the time needed to finish the simulation rises to over half an hour. In view of the tradeoff between grid size and simulation runtime, the grid size is therefore set to ten. A grid size of ten is equivalent of having 100 agents in the organization.

Similar results are obtained when varying the number of belief dimensions for the agents in the organization and the environment. Figure 20 illustrates the influence of the belief vector size on average equilibrium knowledge and steps until equilibrium for 5, 10, 30, 50, 75, and 100 belief dimensions. In general, the fewer beliefs an agent or the environment holds, the faster equilibrium is achieved, and the higher the average individual knowledge. Again, an exception has to be made for small belief size vectors, when the number of dimensions is less than ten. Here, the detoriation of knowledge is caused by a crowding-out effect. Due to the learning process, superior beliefs may be replaced, therefore influencing knowledge levels negatively. Again, like for the grid size, larger belief vectors exponentially increase the computation time of the simulation. Therefore, a belief size of 75 is chosen to balance the trade-off between accuracy and computation time.

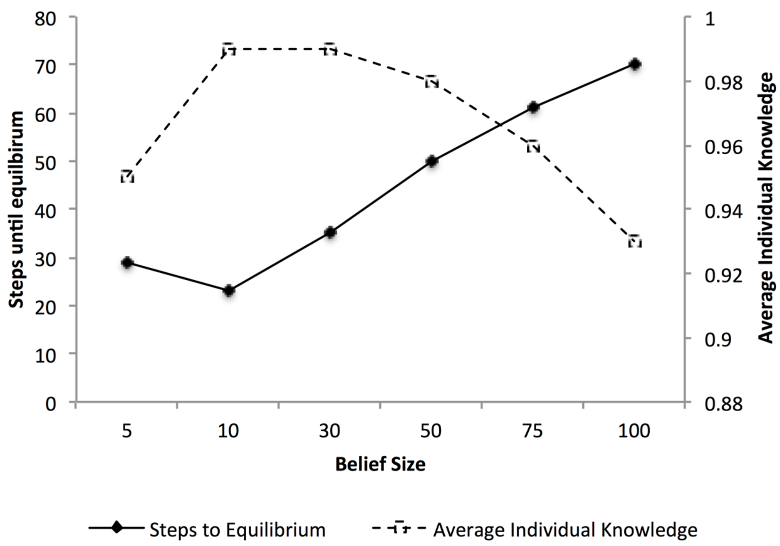


Figure 20: Calibration for belief vector size holding grid size (n=100) fixed

Basic Parameters for the Interpersonal Learning Model

With the conducted calibration for number of iterations, length of the simulation, grid size, and belief size, a set of initial parameters ensuring reasonable results have been identified. Table 12 states the parameters on which variations were conducted, and the values chosen for the simulation are underlined.

Table 12: Overview of the initial simulation parameters

Parameter	Variation	Remark
grid size	2, 5, <u>10</u>, 15, 20, 50	Number of consecutive rounds
individual beliefs	5, 10, 30, 50, <u>75</u>, 100	Number of repeated simulation runs
iterations	50, 100, 150, <u>300</u>, 1000, 10000	Length and width of the grid
time steps	<u>300</u>	Number of beliefs the reality and agents are holding

After the formal model has been transferred into computer code it is used to conduct virtual experiments for answering the research questions. The next chapter describes the structure of the experiments, performs the experiments, and analyzes the results.

6. Simulation Experiments

This chapter contains simulation experiments to answer the questions, how the logic of unlocking can be integrated into the path dependence framework and, how means like reconfiguration, turnover, and a top management team influence the probability of unlocking organizational paths. Before the experiments are conducted, a simulation model depicting the three-phase path formation process serves as a baseline. Experiments are then performed through extending the baseline model. In the first experiment, an exogenous environmental shock, as the triggering event for unlocking, is included. If organizations can escape the lock-in, it is concluded that the extended model integrates unlocking and depicts a four-phase framework of path dependence. After assuring that the model captures unlocking, extensions to answer the second part of the research question are included. While the second experiment captures turnover and reconfiguration, the third experiment captures the influence of a top management team on unlocking.

6.1 The Baseline Model: Path Formation

The baseline model depicts a path formation process for the different similarity values. The results of the baseline model not only have to show that an organizations is locked into a stable equilibrium, but also that heterogeneity, necessary for endogenously unlocking paths, is maintained during the formation process. To comprehend the influence of the similarity parameter on heterogeneity only this parameter is altered, while all other parameters are kept fixed during the simulation run. Through this procedure, changes in the results of the simulation are directly linked to the similarity based selection and learning mechanism. To get a fundamental understanding of the basic simulation model and why the performance varies with the similarity parameter, the average individual knowledge is measured for all similarity values and the results are outlined in Figure 21.

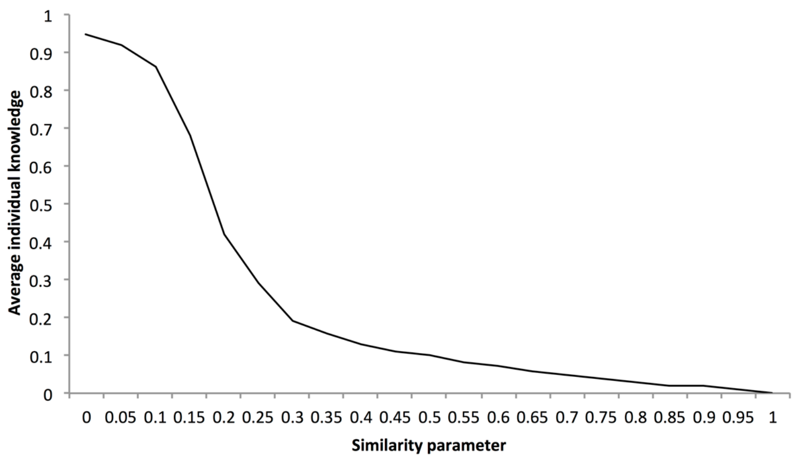


Figure 21: Average individual knowledge for similarity values between 0 and 1

Figure 21 shows the importance of the first six similarity values, ranging from 0 to 0.25, as they exhibit a great impact on the change in the gradient of the curve. For similarity values greater than 0.3, the gradient only changes infinitesimal and further increasing the similarity value has little impact on the average individual knowledge. Constraining the similarity values for further experiments to a small subset, ranging from 0 to 0.25, therefore results in higher clarity and the time needed to conduct the experiments is significantly lowered. As the average individual knowledge is declining with an increasing similarity parameter, an explanation for this effect needs to be found, in order to understand the mechanics of the simulation model. Digging deeper into the spatial structure of the organization and the temporal progression of the simulation may therefore be helpful.

6.1.1 Analysis of the Baseline Model

The main finding in the baseline model is that the average individual knowledge decreases with an increasing similarity value. Until a similarity value of 0.1 only slightly and from there on rapidly, until eventually dropping to an average individual knowledge of zero for a similarity value of one. The explanation for this behavior is that with an increase in the similarity parameter, individuals are also increasingly restricted to learn from like-minded agents in their neighborhood and are therefore not able or willing to tap into the knowledge of superior agents. This is in line with the concept of cognitive distance (Nooteboom, 1992) or individual absorptive capacity (W. D. Cohen & Levinthal, 1990), which says that prior experience impedes individuals to absorb distant knowledge bases without much effort. The similarity parameter hence reflects how difficult it is for individuals to tap into unknown knowledge domains. Darr and Kurtzberg (2000) emphasize that with increasing similarity between individuals, the effectiveness of search for new knowledge and the adoption of knowledge increases as well. The results of the simulation confirm Darr and Kutzberg's results, as they show that an increase in the similarity parameter for selecting other individuals to learn from also hampers the effectiveness of organizational learning, decreasing the average individual knowledge. At the same time, a decrease in average individual knowledge means that agents must, to some extent, disagree with the environment vector. This leads to the question of whether all agents disagree with the environment to the same extent or if beliefs of agents differ from each other. For a deeper understanding of the belief distribution within an organization, a look at the spatial structure of the cellular grid proves helpful. In Figure 22, the cellular automata grid, as a representation for the structure of an organization, is depicted for exemplary single simulation runs. The grid has a width and height of ten cells, resulting in a total of one hundred cells. Each number in a cell represents an individual agent, and agents with an identical set of beliefs are assigned the same number. In this context, neighboring agents with the same number and therefore the same beliefs are assumed to belong to the same group. A group is in the grid defined as an entity consisting of one or more agents sharing the same beliefs, but differing from adjacent agents at least on one dimension. Analogies may be drawn to organizational units like sales, finance, manufacturing, or marketing departments. Each unit comprises individuals with different skills and educational backgrounds, making an organization heterogeneous. Furthermore, groups may differ in their composition with regard to the number of agents within the group and number of agents connected to agents of other groups. Agents in different groups may also differ in how much the belief sets correspond with each other. For example, engineers working in manufacturing are probably more different compared to marketers, as compared to engineers in R&D.

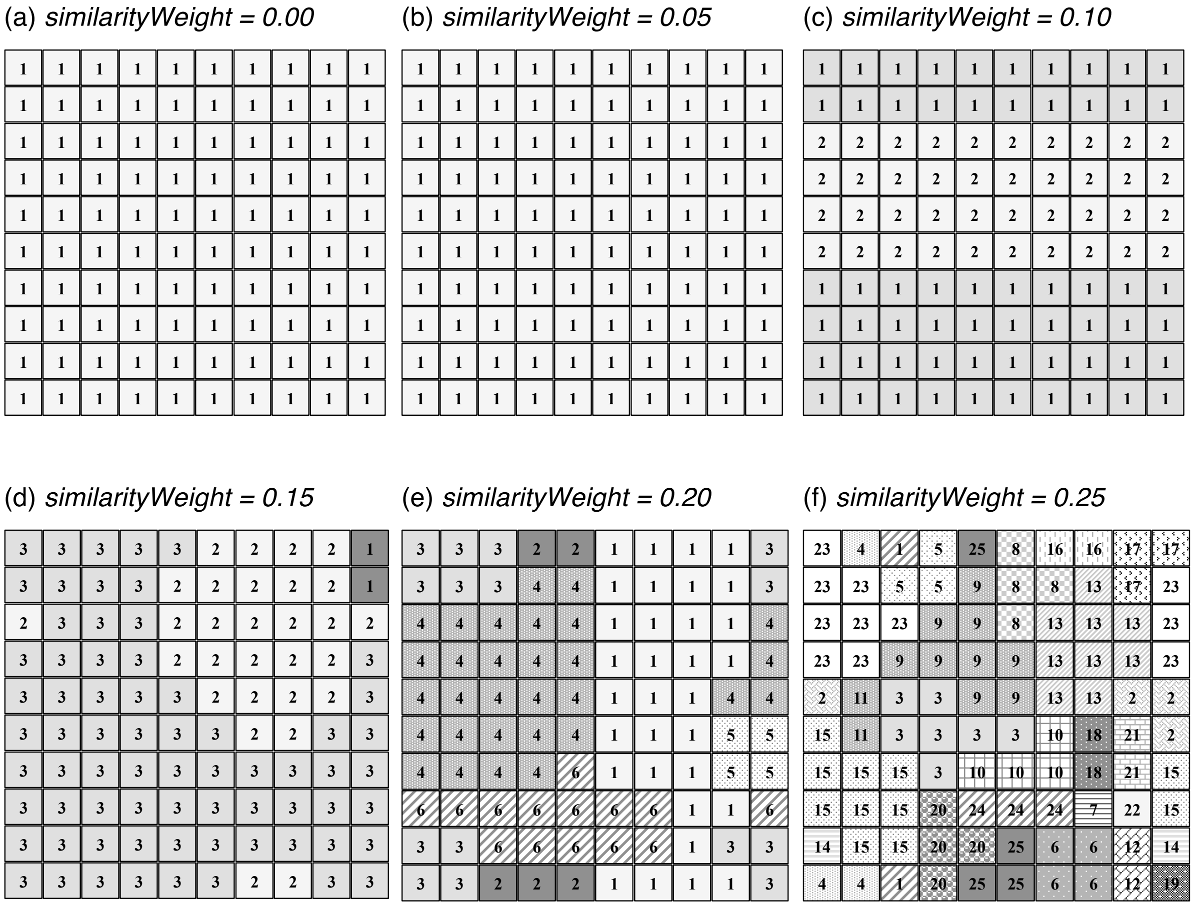


Figure 22: Emerging group structure for *similarityWeight* values ranging from 0.0 to 0.25

In the case where the similarity value equals 0 or 0.05, all agents in the baseline model share the same set of beliefs and therefore belong to the same group. A homogenous organizational structure emerges, where agents have indirect access to other beliefs through their neighbors as information flows unrestrained within the organization. For 0 and 0.05, the baseline model reflects the results of the interpersonal learning model of K. D. Miller, et al. (2006). Like in their interpersonal organizational learning model, no disagreement on the dimensions of the environment exists among the agents within the organization. These results also match the findings of March (1991), where all agents share the same beliefs at the end of a simulation run. In both models, the equilibrium is characterized through a lock-in where all agents share the same opinion about the environment and therefore unlocking, without inducing new beliefs in the organization, is impossible. More interesting, and different to existing organizational learning models, from a similarity value of 0.05 and upwards, consensus disappears and multiple groups, holding divergent sets of beliefs, emerge. Now, agents also conform to similar peers instead of only imitating the agent with the highest performance. This behavior leads to the emergence of distinct groups within the organization. In a constant environment, these groups are stable and remain unchanged in composition after an equilibrium state is achieved. Such a fixed and rigid organizational structure is characteristic for a lock-in (Hite & Hesterly, 2001). Also, as the performance of an organization is defined through the match between the belief sets of agents and the environment, the presence of two or more groups implies, in most cases, superiority of one group over other groups, because the belief vectors differ. This explains the differences in average individual knowledge mentioned above. Furthermore, the stability of the average individual knowledge and number of groups implies that agents learn from neighbors of the same group, despite the possible superior knowledge of adjacent agents. Therefore, even in the case of multiple groups, the rigidity of the organizational structure in the equilibrium affirms the notion of lock-in on organizational level. But, unlike Arthur's urn model of path dependence, yet, in accordance with the three-phase model of organizational path dependence from Sydow, et al. (2009), variation in agent behavior is still possible on group level. Since in each round the selection algorithm searches the agent's immediate neighborhood for better, or at least equally good, performing agents, and randomly selects one of them for updating beliefs, switching learning partners may also happen in a stable equilibrium. Indeed, observing changes in interactions shows that agents are not updating their beliefs from only one agent, but from a variety of neighbors within the group.[36]

[36] A measurement parameter named *switchingInteraction* counts after each run how many times switching of learning partners took place.

Considering, that according to Vergne and Durand (2010) paths can only be unlocked if an exogenous shock occurs, it is important to note that variation alone does not suffice to unlock paths. Different to prior modeling efforts, like for example Peterman, et al. (2012), the simulation model reflects this assumption by allowing for variation, but at the same time highlights that unlocking cannot occur solely through this variation. Specifically, agents may locally switch their communication partner without influencing lock-in on the macro level. Furthermore, the higher the similarity parameter, the more groups emerge. Here, the similarity based selection mechanism causes a separation effect in the organization that leads to homogenous beliefs within a subunit, but heterogeneous beliefs across subunits. While the emergence of groups can be beneficial to the diversity of beliefs in the organization, it is, as shown, detrimental to the average individual knowledge, and therefore to firm performance. Figure 23 shows the correlation between number of groups and belief diversity in the organization. Both, belief diversity and number of groups, are a measurement for heterogeneity within an organization (see D. A. Harrison and Klein (2007) for a description of different heterogeneity measures).

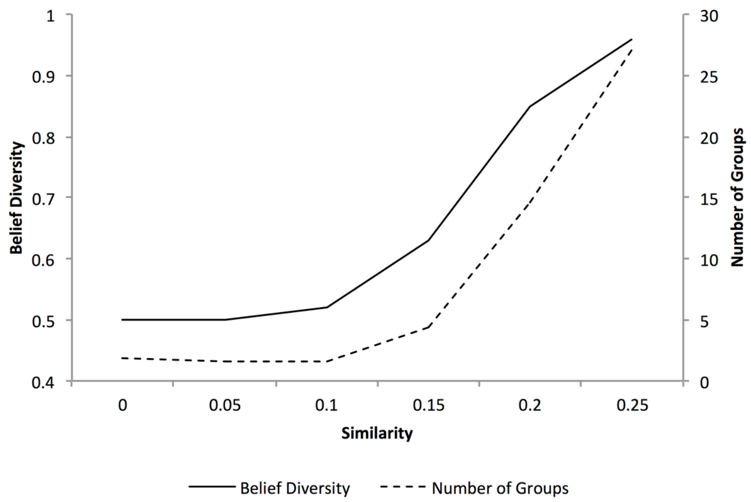


Figure 23: Number of groups and belief diversity over similarity values

As mentioned in Chapter 2, heterogeneity is a necessary condition for unlocking organizational paths. Without heterogeneity a broadening in the scope of options in the lock-in phase is not possible. Therefore, it may be assumed that organizations exhibiting a higher degree of belief diversity can unlock paths more easily. More interesting insights into the behavior of the simulation model and the accompanying path formation process may be revealed, by turning from the spatial dimension to the temporal dimension. Initially the average individual knowledge amounts to zero in the path formation process. This is, because at the beginning of the simulation run, a third of the agents in the organization are, on average, right about the state of an environmental dimension, and another third is wrong. The remaining third has no opinion or knowledge about the environment. Because agents in the organization learn from their peers as the simulation progresses, the average individual knowledge is expected to increase over time, until a stable equilibrium is achieved. While the aggregated results suggest a constant gradient, because of the properties of Monte Carlo simulations (see Mooney (1997)), the situation might be different for single runs. Therefore, Figure 24 shows the progression over time for four independent single runs, instead of the aggregated results.

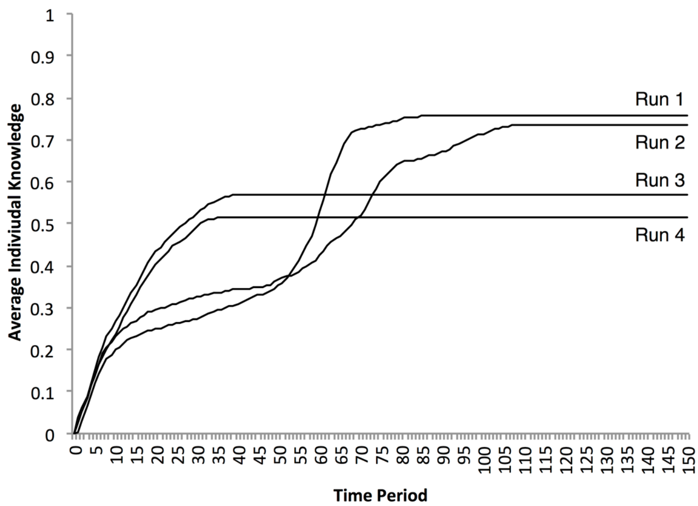


Figure 24: Average individual knowledge over time for random single runs and a similarity value = 0.2 exhibiting changing (run 1 & 2) and constant gradients (run 3 & 4)

Surprisingly, in some simulation runs, phases of rapid growth are disrupted by phases of slow growth or near-stagnation (run 1 & 2), while in other runs the growth remains relatively constant until lock-in is achieved (run 3 & 4). The development of run 1 and run 2 bears analogies to the previously mentioned Setterfield-Type of path dependent evolution described by Martin and Sunley (2010), where a temporal meta-stable equilibrium is disrupted by endogenous change. But, special notice should be given not to confuse such a meta-stable equilibrium, or phases of slow growth, with lock-in. Lock-in is a state which cannot be escaped without the presence of tremendous exogenous pressure for change (see Vergne and Durand (2010)), while in the case of a meta-stable equilibrium, the system may change endogenously (see Romanelli and Tushman (1994)). In the simulation model the similarity based selection algorithm, and depending on that, the learning rate, give explanations for the dissolution of such meta-stable equilibriums. As agents select a learning partner based on knowledge and similarity, it might be that agents with the highest average individual knowledge are merely underrepresented in the learning sample and agents learn from more similar agents. Because learning increases similarity, the probability of learning from superior agents further decreases. As knowledge is not absorbed through agents in the organization, the growth of average individual knowledge may slow down. But, when knowledge from superior agents is eventually absorbed, it may trigger a chain reaction. Then, knowledge rapidly diffuses within the group or organization, initiating a phase of rapid growth in knowledge. Even more interesting is, that for runs with alternating phases of slow and fast growth (runs 1 & 2), the average individual knowledge is higher as for organizations with one steady growth phase (runs 3 & 4). This result reflects the finding of March (1991) that slow learning will generate a higher knowledge level, but at the same time take longer to achieve a stable equilibrium, and that fast learning will result in a lower knowledge level, but achieving a stable equilibrium earlier.

Having a look at the spatial and temporal dimension of the simulation model helped to obtain an understanding of the mechanisms in the model. Now, the results will be discussed with regard to the concept of organizational path dependence.

6.1.2 Discussion

In contrast to the original interpersonal learning model, the model at hand depicts the emergence of groups, and therefore heterogeneity, due to the similarity value in the selection and learning mechanism. As previously argued, agents in organizations do not only learn from the best performing agents in the organization, but also learn from similar agents. Observing the learning process over time shows that it still resembles a three-phase path formation process. At the beginning, the diversity of beliefs has to be great enough to allow the organization to adapt towards different environments (*Phase 1*). Hence, the scope of options is nearly unrestricted and allows for different historical trajectories. Figure 25 shows the diversity of beliefs within the organization over the simulation time. In the first phase of path formation the belief diversity has to be one, in order to allow the organization to adapt towards all possible states of the environment. In Figure 25, the belief diversity at t=0 equals one, meaning that the organization can adapt to all possible states of the environment. Once the selection, and subsequent self-reinforcing learning process, takes place, this variety narrows down (*Phase 2)* until the organization eventually locks into one out of multiple possible stable equilibriums, which cannot be escaped endogenously (*Phase 3).* Figure 25 shows the path formation process over time for the different similarity values and distinguishes the three phases from each other. Depending on the similarity value, Phases 2 and 3 are longer or shorter, but lock-in is still achieved. The distribution of time to lock-in over the similarity values is inverse u-shaped, with a maximum at *similarityWeight = 0.2* and a minimum at *similarityWeight = 0.25*. Hence, the organizational learning process is shortest for *similarityWeight = 0.25*. In the basic interpersonal learning model, with a *similarityWeight = 0,* only one set of beliefs is reproduced (*diversity = 0.5)*. This finding matches the results of March (1991) and K.D. Miller et al. (2006). Different to that, in a model including non-zero similarity parameters, more than one set of beliefs may be reproduced (*diversity > 0.5)* as different groups emerge. As within each group potentially a different set of beliefs is reproduced, heterogeneity is partly preserved over the simulation run. Especially for *similarityWeight > 0.10* the diversity of beliefs is maintained. Still, even though heterogeneity is present in the lock-in, the learning pattern would be reproduced infinitely without pressure to change from outside of the organization (Martin & Sunley, 2010). Therefore, it can be convincingly argued that the model resembles a path formation process, while at the same time allowing for the emergence of heterogeneity through groups. As already laid out, heterogeneity alone is not sufficient for unlocking organizational paths, as it needs an exogenous shock that potentially triggers a reaction (Vergne & Durand, 2010). The first set of experiments includes an exogenous shock in the environment and considers how paths may be unlocked through exogenous pressure.

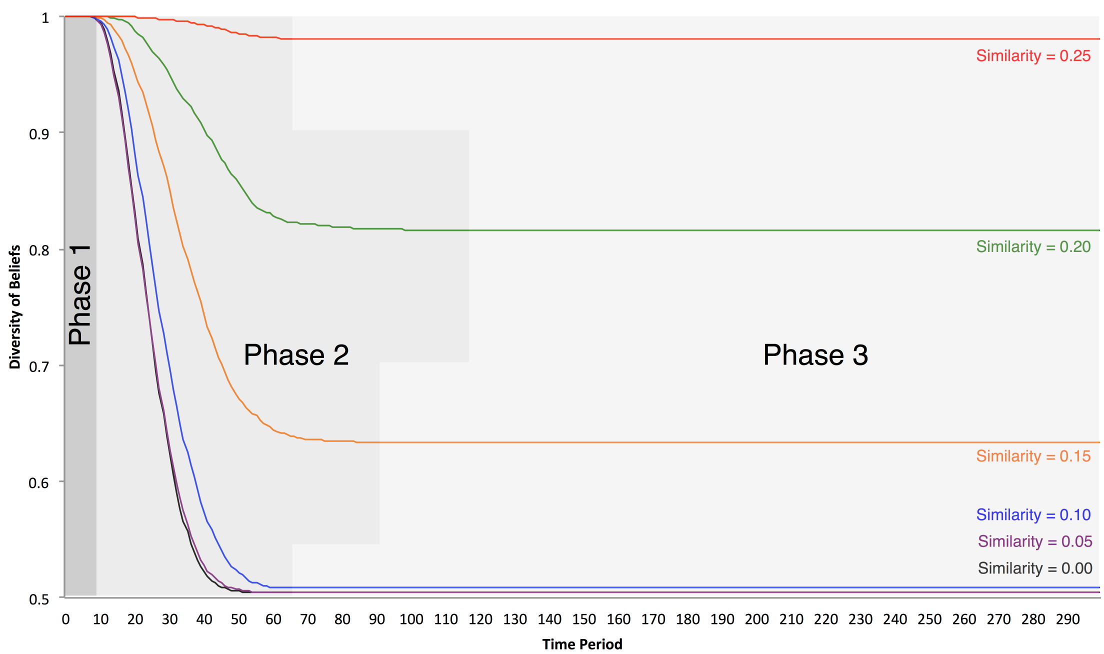


Figure 25: Three phases of path formation in the baseline model

6.2 First Set of Experiments: Inducing an Exogenous Shock

The baseline model resembles the three-phase path formation process from Sydow et al. (2009), but, at the same time, allows for heterogeneity in the lock-in phase through the emergence of a group structure. In this section, a series of experiments is addressing the exogenous shock, or punctuation, that are, according to Vergne and Durand (2010), necessary to unlock organizational paths, and potentially extends the three-phase model to a four-phase model of path dependence. To account for an environmental exogenous shock in the simulation model, the environment vector is replaced with a randomly generated vector at time step 150. This value was chosen to allow enough time for lock-in to be achieved prior to the exogenous shock, and sufficient time thereafter, to enable the organization to recover from the shock. With this setting, the average individual knowledge and the number of groups are observed for the six similarity values. This helps to gather a better understanding of the extended model and to figure out how the model depicts a four-phase path unlocking process. In Figure 26, the results over the different similarity values are shown and compared to the baseline simulation model in the previous chapter.

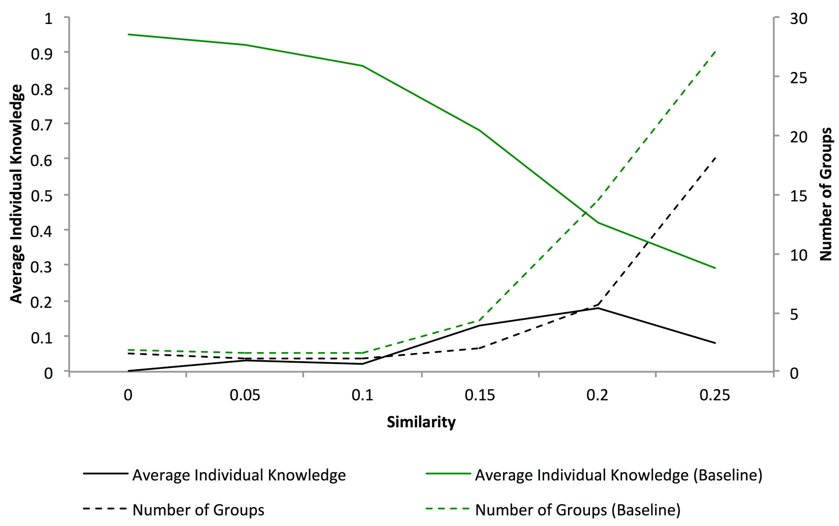


Figure 26: Influence of an exogenous shock in the environment on average individual knowledge and number of groups

What immediately stands out is that the results confirm the detrimental effects, found by March (1991) and Seidel (2012), of environmental change on the average individual knowledge. The negative consequences are especially apparent for the case of *similarityWeight = 0,* where the average individual knowledge over all iterations drops from 0.95 to 0. This result is not surprising, as prior to the shock agents in the organization shared the same beliefs over all dimensions and belief diversity equaled zero. According to Ashby's law of requisite variety, the absence of diversity in beliefs impedes the adaptation towards a new environment (Ashby, 1956). Therefore, the organization is not able to recover from the exogenous shock.

For all other similarity values, the average individual knowledge exceeds zero after the shock occurred and rises to a maximum for *similarityWeight = 0.2.* For similarity values bigger than 0.2, the average individual knowledge is again declining. As in the baseline model, with a static environment, the relationship between similarity and average individual knowledge is inverse u-shaped in the presence of an exogenous shock. Comparing the number of groups, at the end of each run, for the extended model with the number of groups of the baseline model, it can be recognized that the number of groups diminishes (Figure 26). As the number of groups is directly related to the diversity of beliefs within the organization, it can be concluded that endogenous adaptation towards the new environment reduces heterogeneity. With this, the *similarityWeight* parameter extends the internal variety argument of Kim and Rhee (2009), by proposing a mean how variety emerges within an organization independent of external beliefs. To further emphasize the positive effects of heterogeneity on organizational change, Figure 27 depicts the *change in structure*, as measured by the proportion of iterations where the organizational structure is modified over the similarity values. The modification of the organizational structure is defined as in how many runs the structure of the groups is altered. Here, the organizational structure refers to the size, number, and geometrical shape of groups in the grid.

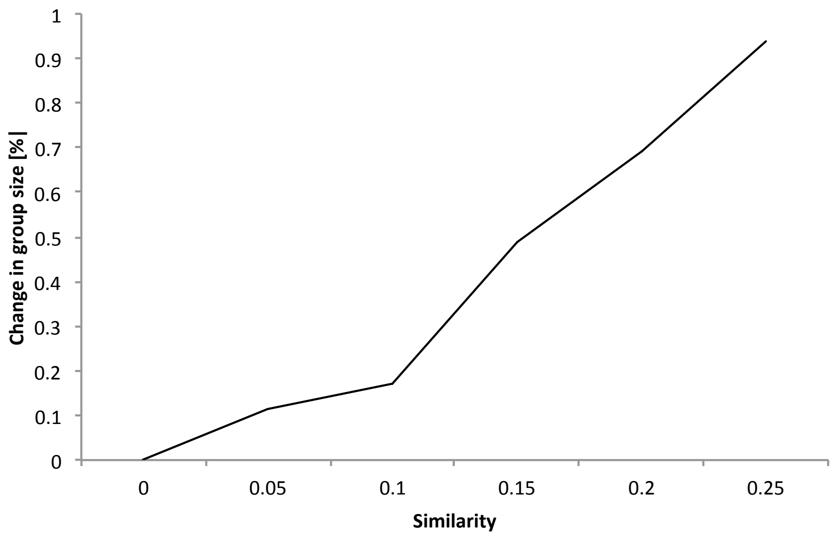


Figure 27: Changes in organizational structure in dependence of similarity value

Again, the results confirm prior findings that heterogeneity empowers an organization to change (see for example T. Kim and Rhee (2009), March (1981), Levinthal (1991) or Hambrick, et al. (1996)). While for a *similarityWeight = 0,* no changes in the organizational structure are observed, for values greater than zero the situation is different. Especially for values greater than 0.15, the organizational structure changes in over 50% of the simulation runs and for *similarityWeight = 0.25* the percentage approaches one. It can therefore be concluded that, through changes in an exogenous environment, organizational change is initiated through the same selection and learning process, which lead to lock-in in the first place. But, while organizational change is a necessary condition for a path to be considered unlocked, it is not sufficient. Change must also lead to a higher average knowledge compared to the situation after the exogenous shock to be considered as a "*real choice*" for an organization (Arthur, 1994; Sydow, et al., 2009). But, it should also be recalled that unlocking may come at a cost and therefore affects firm performance negatively (Vergne & Durand, 2010). So even if the performance does not approach the performance values before the shock occurred, an organizational path may still be considered unlocked, because the organization changed and adapted to the new environment. To sum it up, it is safe to assume that unlocking occurs if after an exogenous shock the average individual knowledge exceeds a certain threshold performance value. This value might be lower as the pre-shock performance value, but higher compared to the performance in the period the shock occurred. To define the threshold performance value, Figure 28 exemplarily exhibits the probability of unlocking for three different threshold values. The graph shows that qualitatively the results do not change for different threshold values. For all three threshold values the probability of unlocking increases until a similarity value of 0.2 and then decreases again for greater similarity values. As the qualitative results remain stable over the similarity value, for further experiments a threshold of 0.1 is chosen.

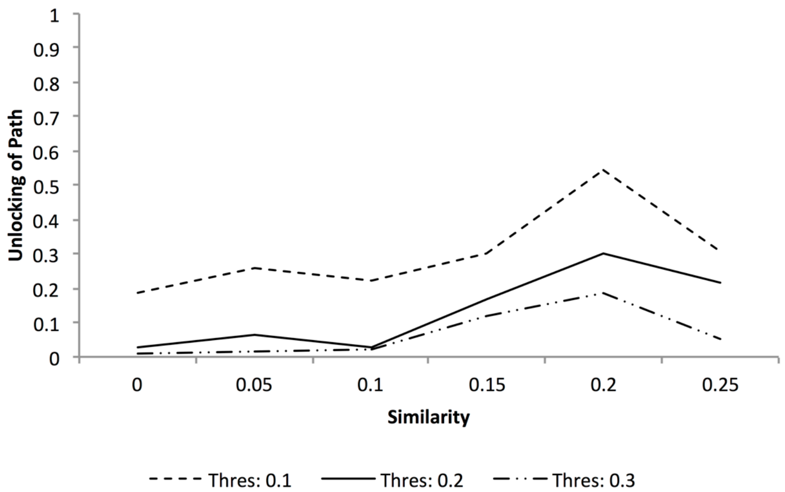


Figure 28: Probability of unlocking for different threshold and similarity values

Although the organizational structure does not change for a similarity value of zero, unlocking of paths occurs in 18% of simulation runs. This contradicting behavior can be explained by the properties of the exogenous shock in the simulation model. As the new environment is randomly determined in period 150, it can occur that the old and new environment match to some extent, and the average individual knowledge therefore does not drop to zero when this happens.[37] The reason why the average individual knowledge, over all iterations, after a shock still drops to zero is explained by the properties of how knowledge is computed. If an agent matches the environment on none of its dimensions, and does not possess any zeros in its belief vector, the individual knowledge equals -1. But, over all iterations, the negative knowledge values are offset by positive knowledge values, causing the average individual knowledge to equal zero.

6.2.1 Discussion

The results of the first set of experiments extend the three-phase model of path dependence to a four-phase model of path dependence by including an exogenous shock. As the selection and learning process can retain heterogeneity, depending on the similarity value, an exogenous shock is a trigger to overcome lock-in and to adapt towards the new environment. Because the unlocking of paths is not initiated through a purposeful action, but by the same mechanisms that lead to the lock-in, it can be interpreted as *path dissolution*.

The results also show that heterogeneity is a necessary condition in order to effectively unlock paths. With increasing similarity value, and along with that increasing belief diversity, the probability of unlocking paths is increasing until a similarity value of 0.25. For a similarity value of zero, where no heterogeneity is present in the organization, the organizational structure remains unchanged and unlocking can only occur if some dimensions of the environment do not change during the shock. Escaping an organizational path where all dimensions change, and without inducing beliefs from outside the organization, is not possible. But, the results also call attention to the adverse side of too much heterogeneity. If agents posses increasingly diverse belief sets they are restricted in their learning capability, for similarity values greater than zero. In this case, heterogeneity, emerging through the learning process, hampers adaptation and eventually has a

[37] While single cases could be observed where the performance increased after the shock, the average behaviour shows a significant drop.

detrimental effect on average individual knowledge, which is seen best for a similarity value of 0.25. Hence, an organization has to find the optimal balance between heterogeneity and average individual knowledge. Prior literature already points out the difficulties organizations face to "*embrace and manage*" (D.A. Harrison & Klein, 2007: 1199) heterogeneity within an organization or single groups.[38] While heterogeneity may stimulate creativity, it can also lead to conflicts, disagreement, paralysis in decision processes, or excessive turnover (K. Y. Williams & O'Reilly, 1998). Besides that, it is the negative effects of heterogeneity on firm performance that impede organizations to embrace heterogeneity. But, the results of the first set of experiments show that only if an organization allows for, and is capable of nurturing, heterogeneous beliefs it can effectively unlock paths. Therefore, to allow for the emergence of heterogeneous groups, organizational niches should not be assessed according to current markets conditions, but have to be protected from market forces. For example, Kemp, et al. (2001) showed, for path dependent technological regimes, that an strategic niche management, covering "*the creation, development, and breakdown of protected spaces for promising technologies*" (Kemp, et al., 2001:270) stimulates co-evolutionary processes necessary to unlock technological paths. To achieve such protection of niches in organizations, subsidies or groups could be semi-isolated (Fang, et al., 2010) or isolated (Galbraith, 1982) from the rest of the organization. Means for organizational isolation of groups are for example the creation of skunk work units (Fosfuri & Ronde, 2009) or internal corporate venture programs (Burgelman, 1983). It seems that isolating organizational units contradicts management literature, which claims that organizations need to support communication and knowledge flow between subsidies in order to facilitate innovations (Tidd & Bessant, 2009). But, the finding that isolation facilitates unlocking is only true if the selection mechanism allow for the reintegration of diverse beliefs after the shock occurred. If a reintegration of beliefs through communication between members of different groups would not take place, meaning that the organizational structure remains unchanged, paths could not be unlocked. Therefore, the findings are not

[38] See D. A. Harrison and Klein (2007) for a literature review diversity in organizations.

contradicting, but complementing current research by highlighting that a phase of isolation, emerging through segregated organizational groups, is followed by a phase of communication and integration of beliefs.

Furthermore, the findings of the simulation experiments also dispenses with the view that outsiders are needed to unlock paths. For example, in the model of March (1991), only personnel turnover is considered to counteract the detrimental effects of environmental change when an organization is locked-in. Instead, the findings of the first set of experiments confirm and follow the argument of Sydow et al. (2005: 25), indicating that the emergent structure does not only restrain, but also enable interactions between agents in the organization. While Sydow et al. (2005) argued that the structure enables an organization to break paths, they also emphasized that they did not dealt explicitly with their concept of a three-phase model of organizational path dependence. The results of the virtual experiments show that the three-phase framework of path dependence can be extended to a four-phase framework, which includes the logic of unlocking. As stated, during the additional fourth phase, unlocking might occur by a belief reallocation process, making use of the heterogeneity present in the organization. In order to show how the path formation and unlocking process evolves over time, the diversity of beliefs for the five different similarity values is shown in Figure 29 over the time period. Until the exogenous shock occurs in period 150 the process mirrors the path formation process discussed in the baseline model. After the shock occurred the belief diversity decreases for all similarity values. The change in belief diversity is evident especially for similarity values of 0.15 and 0.2. Higher similarity values prevent the necessary integration, because the dissimilarity between agents is hampering effective learning. For lower similarity values, the organization does not possess enough diverse beliefs to adapt towards the changed environment. This again emphasizes that too much heterogeneity may hamper unlocking, and therefore organization should actively balance heterogeneity and performance. After the shock occurred, the organization adapts towards the new environment and a new stable equilibrium is achieved again, indefinitely reproduced in the absence of an exogenous shock. Like in the punctuated equilibrium, the process of path formation and unlocking repeats itself until the diversity of beliefs in the organization equals zero.

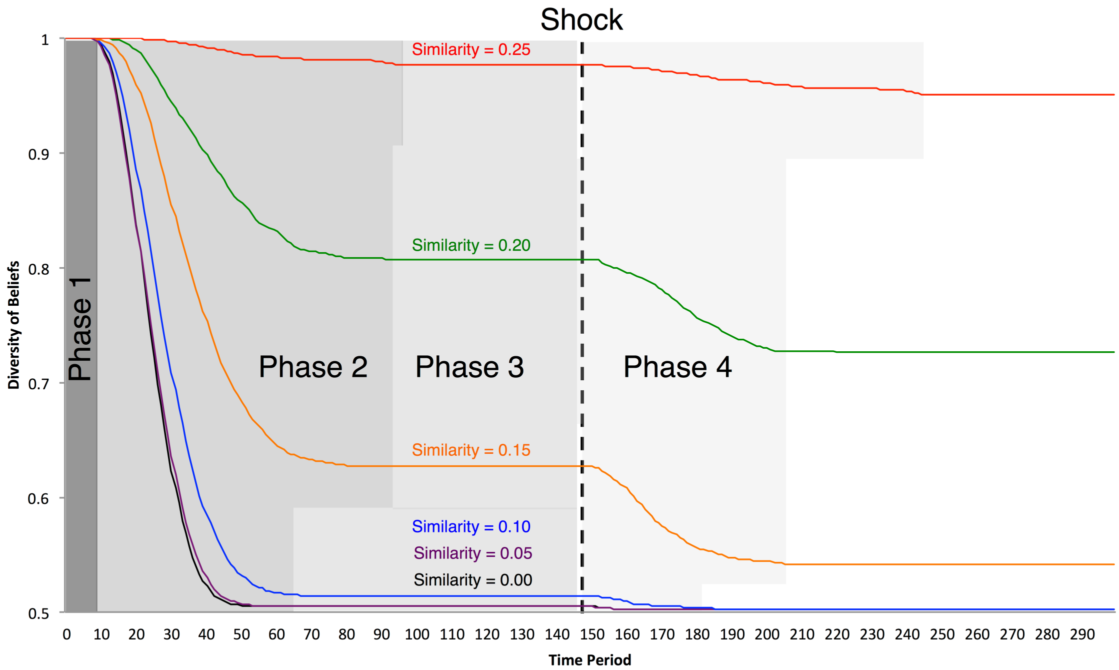


Figure 29: Progression of belief diversity for different similarity values in a four-phase path dependence framework

The findings of the first experiment allow for answering the question on how the logic of unlocking can be included into a stage model of organizational path dependence. Through the similarity based selection and learning process groups emerge that retain heterogeneity through structural separation during the path formation process. Thereafter, the exogenous shock may cause an adaptation process, where the probability that unlocking occurs is depending on the similarity value of the organization. In conclusion, the model captures the logic of unlocking and confirms a four-phase framework of path dependence. While these experiments emphasize the importance of heterogeneity for unlocking, questions on how intentional means influence the probability of unlocking remain unanswered. Deliberate means to unlock paths are also described by the term *path breaking* (Sydow, et al., 2009: 702). Therefore, the next experiments focus on how organizations and the associated management can intentionally make use of or induce heterogeneity into an organization. First, the second set of experiments deals with the question of how rotation and turnover of agents influences the chance of unlocking organizational paths. Second, the third set of experiments examines the influence of a top management team on the path formation and unlocking process.

6.3 Second Set of Experiments: Reconfiguration & Turnover

In March's organizational learning model the drop in average individual knowledge, as a consequence of environmental change, could only be counteracted through the exchange of agents (March, 1991). While the first set of experiments already showed that unlocking might also be endogenously triggered through a shock in the environment, it missed to include intentional means for unlocking. For this reason, the second set of experiments includes turnover as a mechanism for deliberately unlocking paths by replacing agents within the organization through new agents from outside of the organization. While turnover induces new beliefs from outside of the organization, there is also the possibility that the organization makes use of the internal belief variety. Therefore, the case of rotating agents within the organization, depicting a reconfiguration[39] process of the organizational structure, is also examined. Reconfiguration, also referred to as restructuring in management literature, is used as a mechanism to adapt to changing environments and to break path dependencies (Zajac & Kraatz, 1993). By reconfiguration, through rotating agents between different groups, knowledge may be transferred between groups, potentially initiating a belief recombination process and adaptation to the environment after the shock. Furthermore, reconfigurations are commonly known as a way to foster innovation (Dougherty, 1992), increase efficiency within an organization (Bowman & Singh, 2007), and for initiating organizational change processes through the recombination of resources (Karim, 2006). The second set of experiments hence attempts to answer the first two sub

[39] Reconfiguration is referred by Karim (2006: 801) as "*management of resources and structure*".

questions of the second research question, namely (a) how turnover, and (b) how reconfiguration affects the probability of unlocking organizational paths. To answer the questions, the simulation model of the first set of experiments has to be extended with parameters for turnover and rotation. The value range of the two parameters has been chosen based on prior literature and simulation studies. Both parameters exhibit a high and a low value. The high turnover value has been selected in accordance with the organizational learning model of March (1991). March has used a value of 10% for turnover of agents in the organization. Using this value, March found that turnover counteracts the negative impact of incremental environmental change. The low turnover parameter is defined with 1% to represent natural fluctuation within the organization. Similar, turnover values have been confirmed by empirical studies in management literature. For example, Terborg and Lee (1984) found mean turnover rates, for sales staff and the management for a large retailer, to be between 2% and 7%. Therefore, assuming turnover rates of 1% (low) and 10% (high) sounds reasonable and are not far off from empirical findings. With regard to the rotation rate of agents within the organization, it can be assumed that internal reconfiguration actions are more profound and so higher rotation probabilities may be achieved compared to turnover (Bowman & Singh, 2007). Empirical evidence that rotation rates are higher than turnover is provided by Campion, et al. (1994). They found a job rotation rate of 44% for the financial unit of a large pharmaceutical company. As poor performance, induced through a shock in the environment, is a trigger for far-reaching reconfiguration activities (Levinthal, 1991), a higher value (rotation rate = 0.9) compared to Campion, et al. (1994) is selected for the high rotation parameter. For a low rotation value 0.1 was chosen, meaning that 10% of the agents switch their location within the organization after the shock in the environment. Although reconfiguration could be a mean to unlock paths, the literature points out that there is also high risk involved, due to unintended consequences (see for example Lavie (2006) or Bowman and Singh (2007)). Therefore, it is not obvious if and how reconfiguration of the organization can unlock paths. Table 13 summarizes and explains the parameters for the second set of experiments.

Table 13: Characteristic of turnover and rotation values

Parameter	Value Range	Remarks
turnoverRate	Low: 0.01 High: 0.1	Probability of an agent being replaced after an exogenous shock occurred
rotateRate	Low: 0.1 High: 0.9	Percentage of agents changing position within the grid

After turnover and rotation parameters were defined, experiments are carried out through altering the parameters and measuring the effect on belief diversity, average individual knowledge, and in particular on the probability of unlocking organizational paths. The next section examines the turnover of agents, while chapter 6.3.2 focuses on the effects of agent rotation.

6.3.1 Influence of Turnover on Unlocking

To examine the influence of turnover on the probability of unlocking, agents are replaced only once after the exogenous shock occurred. In the case of low turnover in total 1 out of 100 agents, and for high turnover 10 out of 100 agents are replaced on average. This approach is different to March (1991), where agents are replaced continuously. But, in the original organizational learning model the environment changes incrementally each period, while in the model at hand the environment is replaced only once in period 150. Therefore, turnover also happens once throughout a simulation run. The impact of turnover on average individual knowledge is then compared to the results of the first set of experiments, which serves as baseline. In the first set of experiments, no intended actions to change the organizational structure were taken, while with turnover new beliefs from outside of the organization are induced. Figure 30 shows the impact of low and high turnover levels on the average individual knowledge. In both cases, for low and high turnover, exchanging agents in the organization has a positive effect on average individual knowledge and unlocking of organizational paths. When compared to the baseline scenario, an organization with turnover also achieves higher average individual knowledge values.

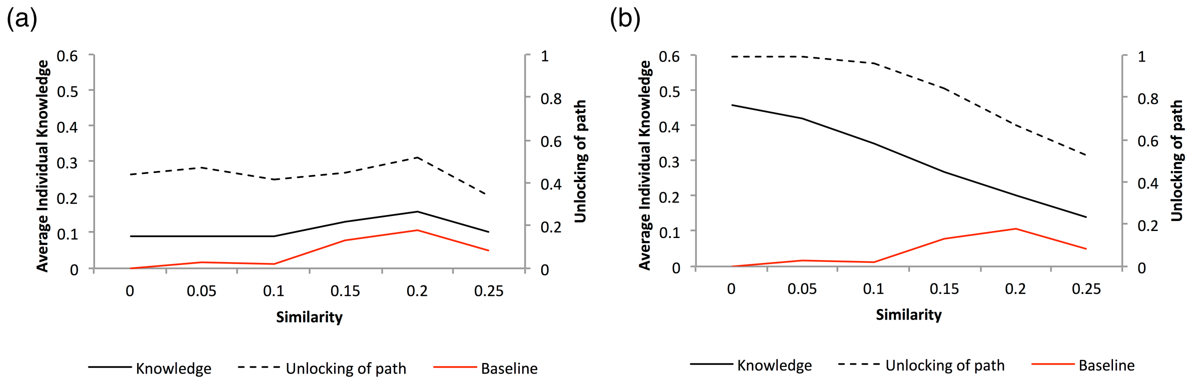


Figure 30: Average individual knowledge and probability of unlocking for (a) low turnover and (b) high turnover values

But, while the low turnover rate only has a small impact on the outcome, the effects of a high turnover rate are more obvious. Like in the original learning model of March (1991), high turnover proves to be an effective mean to counteract the devaluation of knowledge. However, with increasing similarity value, the replacement of agents becomes less effective. A possible explanation is that the new knowledge, coming from outside of the organization through new agents, is not incorporated and newcomers are rather assimilated towards the existing beliefs in the organization. This explanation confirms findings of prior research on organizational socialization. For example, Jones (1983) has shown that socialization of newcomers is not only affected by organizational methods of assimilation, like on-boarding, but also by individual differences. For high turnover values and similarity values greater than 0.1, these individual differences can hamper the assimilation of new beliefs brought by newcomers. While turnover still is effective to unlock paths, the efficiency is dropping with increasing similarity value. This is, because with high similarity values the problem is not necessarily the belief diversity, but rather that new beliefs are not absorbed as high similarity values restrict the learning process to less distant belief sets. Therefore, inducing new beliefs may not facilitate adaptation towards the new environment.

That turnover counteracts the negative consequences of path dependence, and helps to unlock paths, is also in line with prior research in the path literature. For example, Seidel (2012: 128) showed that turnover "*preserves variety indefinitely*" and therefore is helpful to overcome lock-in. But, different to that, the results of the simulation showed that individual traits prevent the adoption of new beliefs. Besides that, as groups emerge through the selection and learning process, the question arises, if agents within groups or agents at the border of groups should be replaced. Agents within groups are only linked to agents of the same group, while agents at the border of groups are linked to at least one agent of another group. Agents at the border of groups may also be comprehended as boundary spanners. Boundary spanners are important for exchanging beliefs between different groups by negotiating between conflicting interests (R. A. Friedman & Podolny, 1992). In this context, boundary spanners are the connective link between two or more groups. Furthermore, boundary spanning is considered to be an explorative activity (Rosenkopf & Nerkar, 2001). Figure 31 compares the results for turnover of agents within and at the border of groups for the low turnover value.

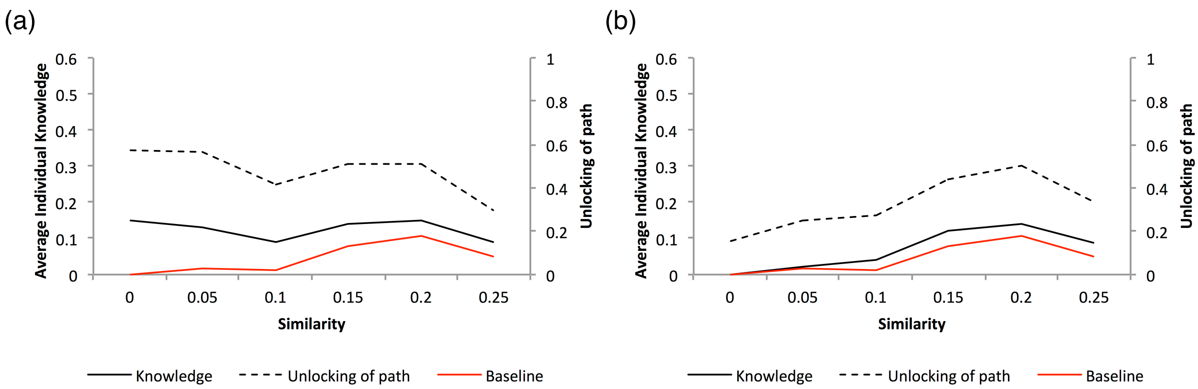


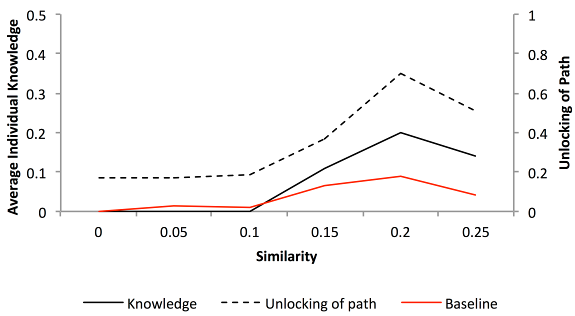
Figure 31: Average individual knowledge and probability of unlocking paths for low turnover (a) within groups or (b) of agents at the border of a group

At first glance, comparing both possibilities suggests that average individual knowledge is higher if organizations replace agents within groups for similarity values smaller than 0.15. But, this impression is deceptive. If the organization consists of one group, for example with a similarity value of zero, obviously no turnover at the border of a group is possible. Thus, the average individual knowledge cannot recover and remains stable at zero. If the organization consists of multiple groups, the difference between the two types of turnover diminishes. For similarity values greater than 0.15, the number of agents being replaced is, on average, the same for both turnover types. Therefore, it can be concluded that the type of turnover has no impact if the same amount of agents get replaced. After examining the impact of turnover on the probability to unlock paths, attention is turned towards a reconfiguration process within the organization.

6.3.2 Influence of Reconfiguration on Unlocking

In case of an organizational crisis, managers may decide to reconfigure the organizational structure in order to improve the ability of the organization to absorb new knowledge (Karim, 2006; Van Den Bosch, et al., 1999). As previously mentioned, the organizational structure is represented through the different groups and the position of agents within the grid. The rotation of agents alters the position of agents in the organization and therefore changes the group composition. Because of this, the rotation of agents is used to depict an organizational reconfiguration process. By interchanging agents between different groups, belief sets that are new to the group are induced. Instead of introducing belief sets new to the whole organization through personnel turnover, rotation only uses the variety already present within the organization. Figure 32 constitutes the impact of low and high rotation rates in dependence of the similarity value on the average individual knowledge and the probability of unlocking.

(a)



(b)

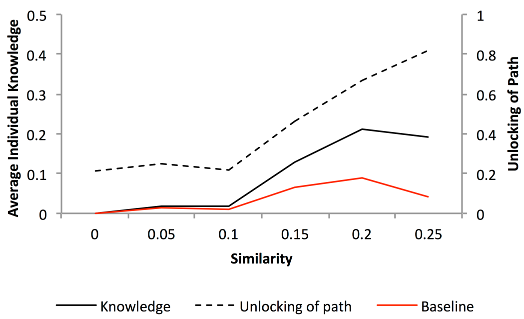


Figure 32: Average individual knowledge and probability of unlocking for (a) low rotation and (b) high rotation values

For similarity values smaller than or equaling 0.10, the rotation probability has only little impact on the average individual knowledge and the probability of unlocking. This can be explained through the absence of belief diversity, best pictured by Figure 22, showing the organizational structure. Again, different belief sets are needed in order to adapt towards a changing environment. But, the rotation of agents does not induce new beliefs in the organization as it makes use of beliefs already existing within the organization. Because belief diversity is low for similarity values smaller than 0.10, adaptation is difficult. The situation is different for similarity values larger than 0.10. Getting back to Figure 32 shows that, for similarity values greater than 0.10, multiple groups emerge and therefore diversity of beliefs, which is positively correlated to the number of groups, exists. Furthermore, in comparison with the results of the first experiment, where an exogenous shock in the environment has occurred, the rotation of agents within the organization leads to a higher average individual knowledge. Also, when compared to turnover, the rotation of agents within the organization seems to be a good instrument to unlock paths for high similarity values. This is in particularly apparent for the similarity value of 0.25 and a high rotation rate, where the probability of unlocking rises from approximately 30% for turnover to over 80% for rotation.

6.3.3 Discussion

Depending on the similarity parameter, turnover and the rotation of agents may prove to be good means for intentionally unlocking paths. While for low similarity values turnover has a great effect on the ability of organizations to unlock, rotation is especially useful for high similarity values. With regard to the four-phase framework of path dependence, Figure 33 and 34 below show the connection between learning over time and diversity of beliefs. For turnover (Figure 33), the diversity of beliefs increases after the shock to nearly one for all similarity values. As time passes, the organization adapts, in particular for low similarity values, and diversity decreases again. Hence, the process almost resembles the prior path formation process, as the organization exhibit high diversity of beliefs through the new beliefs coming from the new agents. This punctuates the findings of March (1991) and Seidel (2012) that turnover can obtain, and therefore prevent a stable equilibrium phase like lock-in, or restore the variety of beliefs within the organization. Hence, turnover facilitates an organization to forget its history and the negative effects associated with the development of knowledge over time.

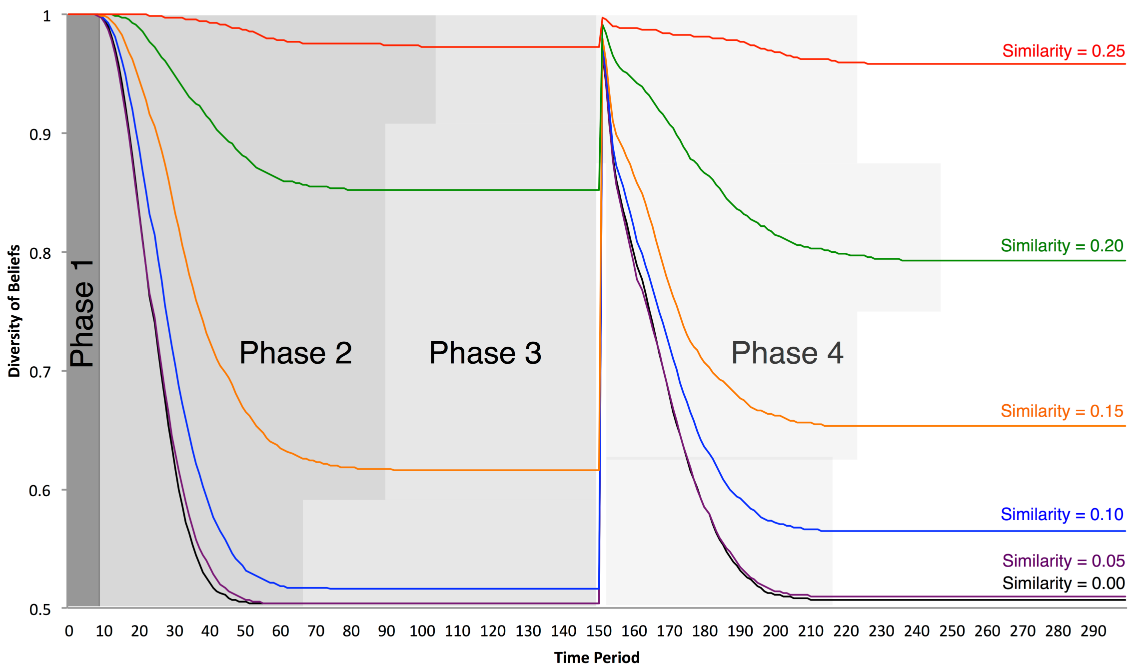


Figure 33: Progression of diversity for high turnover according to the four phase path model

For rotation of agents between organizational groups (Figure 34), changes in belief diversity are greatest for the three highest similarity values. The maximal change is achieved for a similarity value of 0.20. For this similarity value the belief diversity is great enough to allow unlocking, and restriction on learning from others is little enough for the absorption of new beliefs. But, different to turnover, belief diversity is not restored and decreases for rotation over time, hence infinitely unlocking of organizational paths is not possible. Rather, switching the position of agents makes use of existing diversity, instead of inducing new beliefs. The variety of beliefs could also be comprehended as an organizations resource configuration, and resources which are not contributing to adapt towards the environment can be attributed to slack resources (Voss, et al., 2008). After the exogenous shock occurs, these slack resources are reassigned and dissolved due to the rotation of agents. In conclusion, the second set of experiments showed that organizational paths can be unlocked more effectively, compared to the results of the baseline experiments, if an organization actively induces new beliefs into the organization, through turnover or by recombination of existing beliefs through reconfiguration.

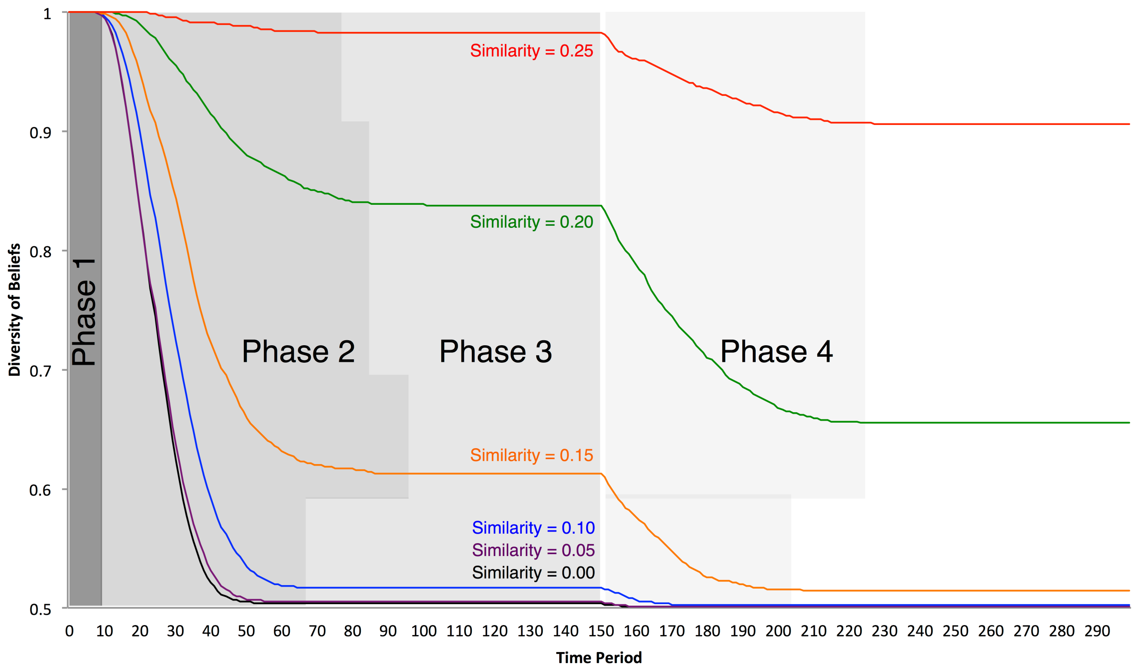


Figure 34: Progression of diversity for high rotation according to the four phase path model

Furthermore, the results hint to the ability of an organization to endogenously unlock paths for high belief diversity values. Endogenously unlocking paths by means that make use of existing beliefs should even be favored over turnover for higher similarity values. But, on the other side, it has also been demonstrated that in the absence of belief heterogeneity, turnover is an important instrument for unlocking paths. Therefore, an organization has to evaluate its internal situation in order to make use of the appropriate mean to unlock paths.

Unlike the baseline experiments, this chapter did consider intentional means for escaping paths, but still missed to view an organization as a hierarchical system. As the findings of Petermann, et al. (2012) showed the importance of a hierarchy on the path formation process, it may also be relevant for the unlocking process.

Hence, in the third set of experiments, the simulation model is extended through a hierarchy, represented by a top management team, that is shaping the beliefs of individuals in the organization.

6.4 Third Set of Experiments: Influence of a Management Team

In the strategic management literature the top management team is regarded as the strategic apex of an organization, responsible for coordinating the behavior of individuals in the organization by aligning it towards the corporate strategy (Mintzberg, 1979; Wiersema & Bantel, 1992). Coordination is defined, according to H. Arrow, et al. (2000), as a mean to achieve agreement between organizational members through generating an interpersonal shared understanding of information and events. Unlike the simulation model of Petermann, et al. (2012), in which the effect of span of control and leading power on the path formation process are emphasized, the inclusion of a top management team is a simple representation of a hierarchy.

6.4.1 Top Management Team Influence on Path Formation

In the model, the top management team agrees on a strategy from which individuals within the organization update their beliefs, regardless if the beliefs of the strategy are wrong or right. Average individual knowledge measurements are then made for the case of low (*p=0.01*) and high (*p=0.1*) influence of the strategy, and therefore the top management, on individuals (Figure 35). To visualize the impact of the top management team, the average individual knowledge is shown over the number of steps in the simulation in the absence of any environmental change. In the case of low influence, the average individual knowledge for similarity values over 0.1 does not achieve a stable equilibrium throughout a simulation run of 300 steps. Despite that, the average individual knowledge improved compared to the situation without a top management team for similarity values of 0.15, 0.2, and 0.25. The effect is especially evident if top management team influence is high, as the average individual knowledge for similarity values greater than 0.1 increases drastically.

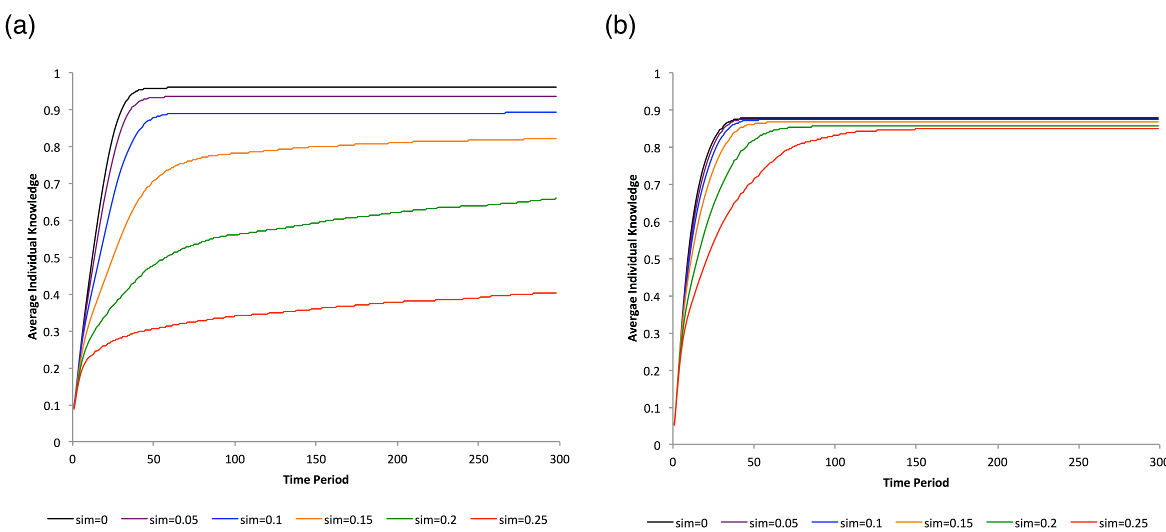


Figure 35: Convergence of average individual knowledge for (a) low and (b) high top management team influence

An explanation for the improvement in average individual knowledge can be found in the coordination effect of the top management on the learning and socialization process of individuals in the organization. As the top management consists of the agents with the highest performance at the beginning of each simulation run, and agents within the organization update their beliefs from their strategy vector, a positive first order effect on the average individual knowledge exists, as agents are assimilated towards the strategy vector over time. Furthermore, a second order effect arises, because agents develop a shared meaning through updating their beliefs from the strategy vector and thus, interpersonal learning is facilitated, in particular for high similarity values. However, for similarity values less or equal 0.1, a slight decrease in average individual knowledge, compared to the standard interpersonal learning model is observed. An explanation of this is provided by the constituting property of the strategy vector. Even if agents within the organization achieve superior knowledge by means of belief recombination, these agents still update their beliefs from the top management team. So, even if an agent has superior knowledge, it will still conform to the top management team and adopt inferior beliefs detrimental to the average equilibrium knowledge. The existence of a top management team also influences the belief diversity within an organization. Figure 36 displays the consequences of a top management team with high influence on the diversity of beliefs within the organization.

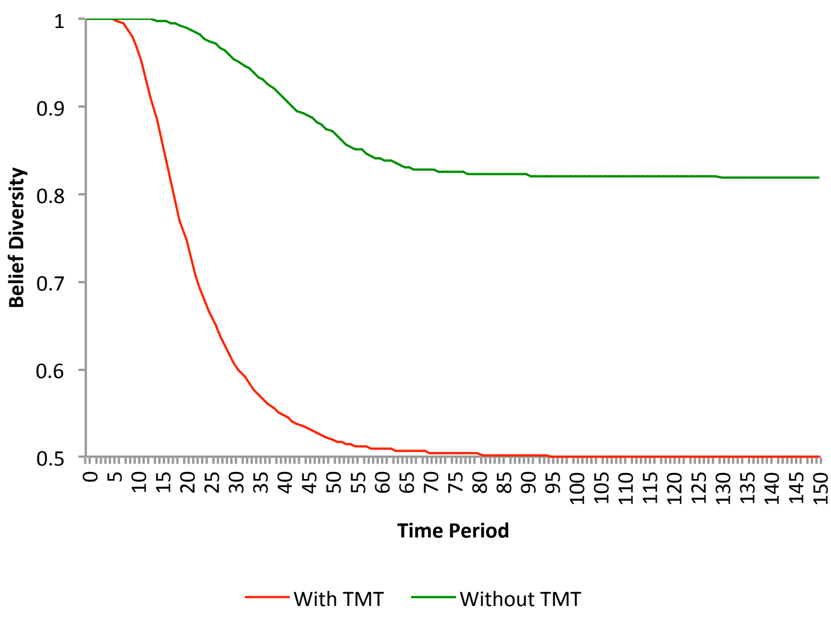


Figure 36: Belief diversity for *similarityWeight = 0.2* with and without top management team influence

Updating from the top management team implicates that diversity decreases and, correlated with that, the average number of groups. As previously shown in the first set of experiments, an organization consisting of agents with homogenous beliefs hampers unlocking, because of the lack in belief diversity. Hence, coordination through a top management team will negatively affect the probability of unlocking. In the next step, the effects of a top management team on unlocking in the presence of an exogenous shock will be examined.

6.4.2 Top Management Team Influence on Unlocking

In this section, the top management is replaced to support unlocking by introducing heterogeneity. In general, a heterogeneous top management team enhances an organization to take action and react to a changing environment (Hambrick, et al., 1996). As mentioned before, the poor firm performance after an exogenous shock puts pressure on the top management members, challenging the current composition of the team. A crisis, triggered by the shock in the environment, can lead to the replacement of top executives (Tushman & Romanelli, 1985; Wiersema & Bantel, 1993). If the forces of change succeed, the level of replacement may vary (Barker III, et al., 2001), and the top management will be exchanged partially or completely. Both possibilities are included in the model through either replacing one of the five top management agents (*low turnover*) or all agents at once (*high turnover*). Unlike before, however, agents in the organization start learning from the top management team after the environmental shock occurred. Still, the top management team is established at the beginning of the simulation and remains fixed until the shock occurs. The effect of the two different turnover rates, for the six similarity values, on unlocking of paths and average individual knowledge is depicted in Figure 37. High turnover of the top management team has a great impact on average individual knowledge and the probability of unlocking.

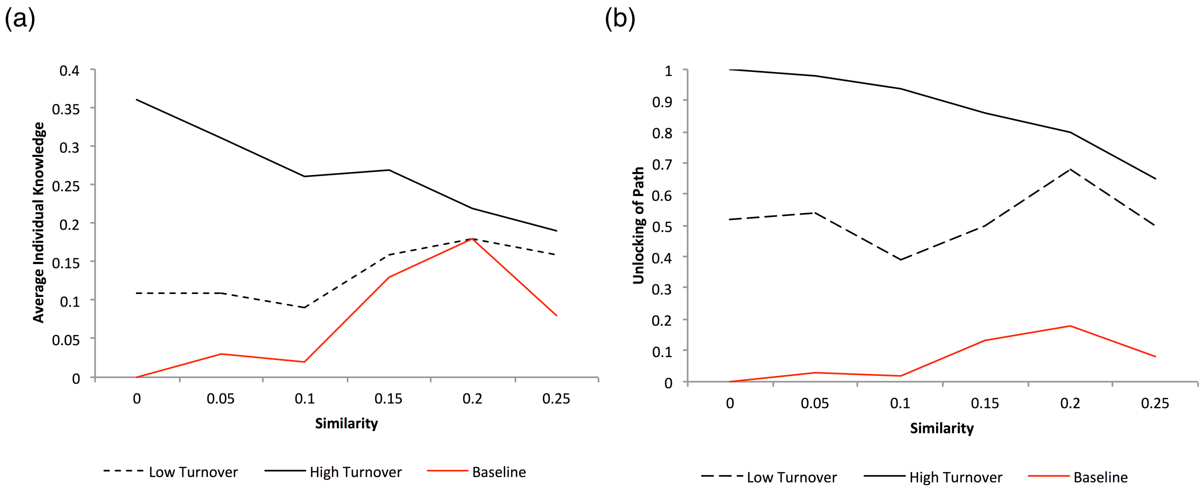


Figure 37: (a) Average individual knowledge and (b) probability of unlocking for low and high top management team turnover and slow updating from the strategy vector

An explanation can be found in the number of new beliefs the new top management agents bring into the organization, and the diffusion of these beliefs to organizational members. But, in contrast to turnover of individuals, top management team turnover is also a good mean for unlocking, when high similarity values are present. This becomes particularly apparent when the influence of the management on agents is high.

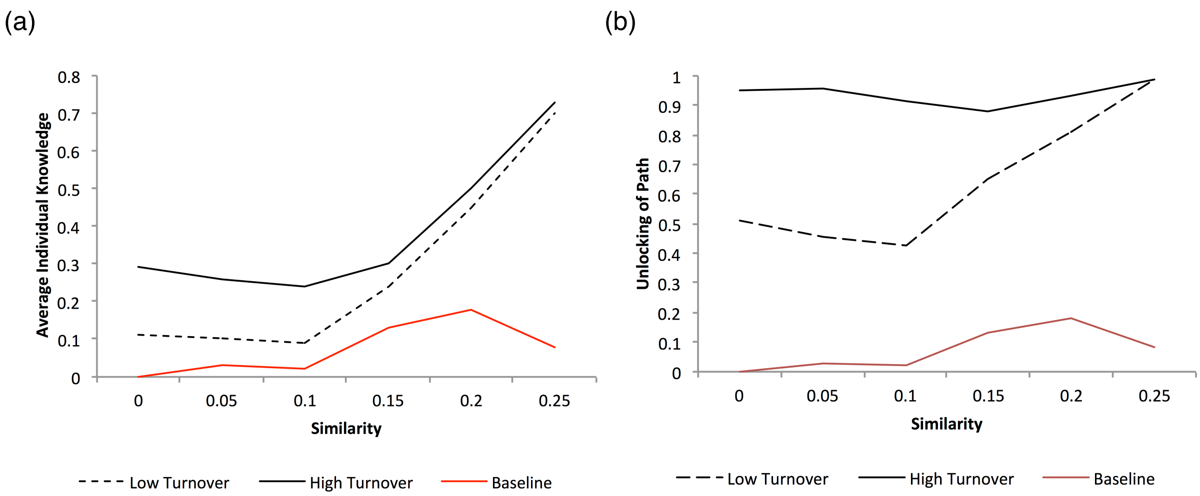


Figure 38: (a) Average individual knowledge and (b) probability of unlocking for low and high top management team turnover and fast updating from the strategy vector

The graphs in Figure 38 show that the difference between high and low turnover diminishes for increasing similarity values. Here, in particular the second order coordination effect has a major influence on unlocking. For example, for *similarityWeight=0.25,* the variety induced through the new top management team members seems to have no great effect on the ability to unlock paths, or the average individual knowledge. Instead, updating from the strategy vector creates a common ground for interactions between agents and enables new interaction structures. Knowledge can be updated between dissimilar agents in the organization with greater ease, and may even help to adapt to the new environment.

6.4.3 Discussion

Top management team influence has a strong impact on the process of unlocking. During the path formation process fast updating from a strategy vector decreases variety, but increases average individual knowledge. On the other hand, when the top management team is formed at the beginning, but agents neglect to update their beliefs from the strategy vector, diversity is preserved (Figure 39).

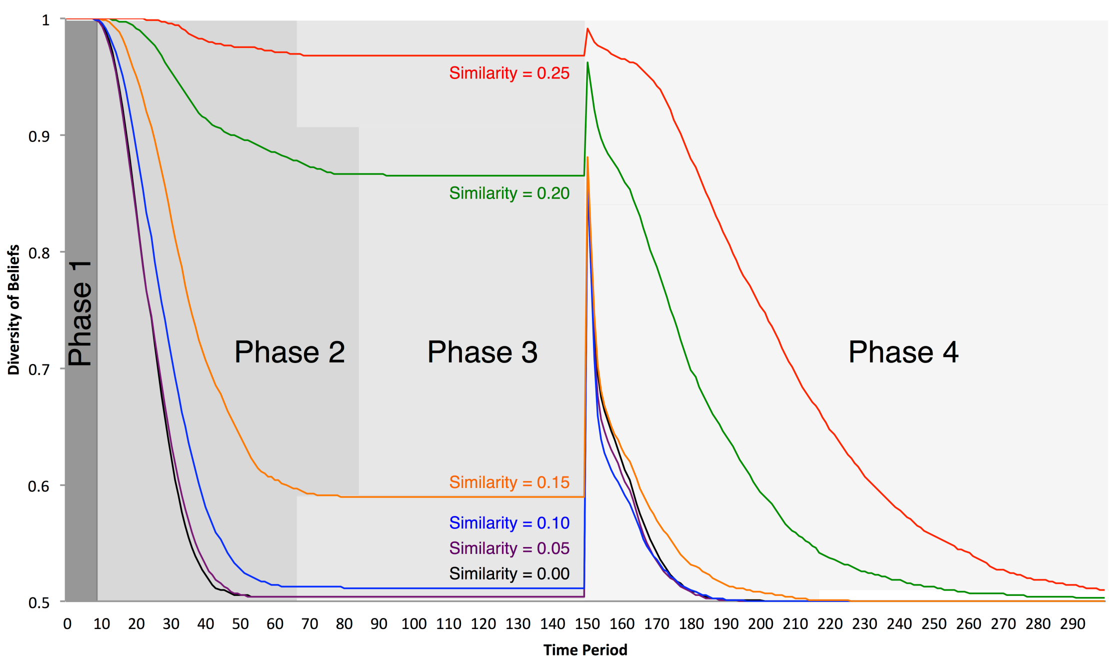


Figure 39: Influence of the top management team on belief diversity for high turnover and fast updating from the strategy vector

Still, the organization will lock into a stable equilibrium until the environment collapses. Here, replacing the top management team can help organizations to unlock paths, because firstly, new management agents induce variety for low similarity values and secondly, learning between heterogeneous agents is facilitated for high similarity values through coordination. In contrast to turnover and rotation, the replacement of top management team members proves to be effective over all similarity values.

6.5 Comparison of the Simulation Outcomes

The series of simulation experiments examined the phenomenon of unlocking organizational paths by integrating the logic into a four-phase model and testing means to unlock paths. The findings show that unlocking according to the four-phase model of Sydow, et al. (2005) is possible, even if no intentional means are applied. Although only one set of options is practiced during the lock-in, leaving the organization trapped into a stable equilibrium, unlocking can be triggered through an exogenous shock. The heterogeneity present within the organization is then used to unlock paths in phase four, as depicted in Figure 6. This also confirms that means that are characterized by heterogeneous agents with diverse beliefs about the state of the environment[40] can enable the unlocking of organizational paths. For intentionally unlocking paths, turnover and rotation of agents were proposed and examined, as they depict the means of invasion and reallocation of resources (Table 4). At last, the influence of a top management team on the unlocking of paths was examined. Thus far, the proposed means were not compared with each other but viewed separately. To answer the second research question of how turnover, reconfiguration, and top management team influence affect the unlocking of organizational paths more thoroughly, this section compares in Table 14 the impact of the different means, contingent on the similarity value. Through that, this discussion extends the findings and shows how a specific mean affects unlocking

[40] These means are listed in Table 4 and comprise cognitive dissonance between agents in the organization, diverging behavior, by-product of path formation or imperfect adaptation.

in relation to other means. Although all proposed means are not mutually exclusive and could be combined, they are here compared independently with regard to the probability of unlocking paths. Further simulation experiments may combine these means and examine interaction effects between them. In Table 14, the most effective mean to unlock paths is highlighted with a dark grey background, and the second most effective is marked in light grey. Noticeable, turnover of the top management team proves to be effective over all similarity values, regardless if the influence of the hierarchy is high or low. According to this result, it is advisable for organizations to exchange the management team in order to escape paths. New beliefs induced into the organization through the replaced management and the communication of a new strategy produces heterogeneity and facilitates the coordination between agents. While for a low similarity value the organizations benefits from the imported beliefs through the new management team members, the coordination effect is the driver of unlocking for a high similarity value. For a low *(similarity = 0.00)* and high *(similarity = 0.25)* similarity value, the second most effective means are turnover and rotation of agents. As organizations with a low similarity value also exhibit low heterogeneity, turnover induces the beliefs necessary to unlock paths. High turnover even ensures the unlocking of organizational paths in the simulation model *(unlocking = 1.0)* for a low similarity value. When heterogeneity is high *(similarity = 0.25),* rotation is the second most effective mean after the replacement of the top management team. As the organization already shows great heterogeneity, importing new beliefs is less effective than making use of uncommitted beliefs. Making existing beliefs accessible to other groups, through the reconfiguration of the organization, proves to be more effective. For a medium similarity value *(similarity = 0.20),* the replacement of the top management team with low influence of the strategy vector is the second most effective mean. Still, at this point, turnover and rotation achieve high values in the unlocking of paths and a combination of different means could obtain a higher unlocking probability. With regard to the second research question, it can be stated that top management team influence has the highest impact on unlocking of paths over all similarity values and that the impact of turnover and rotation on unlocking depends on the similarity value.

Table 14: Comparison of means to unlock paths for low, medium and high similarity values

Similarity = 0.00:

Parameter	Shock	Turnover		Rotation		TMT (0.01)		TMT (0.1)	
		Low	*High*	*Low*	*High*	*Low*	*High*	*Low*	*High*
Unlocking	0.17	0.47	1.00	0.18	0.22	0.52	1.00	0.51	0.95
Diversity	0.50	0.51	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Knowledge	0.00	0.11	0.49	0.00	0.01	0.11	0.36	0.11	0.29

Similarity = 0.20:

Parameter	Shock	Turnover		Rotation		TMT (0.01)		TMT (0.1)	
		Low	*High*	*Low*	*High*	*Low*	*High*	*Low*	*High*
Unlocking	0.53	0.57	0.79	0.67	0.72	0.62	0.8	0.81	0.93
Diversity	0.69	0.75	0.79	0.73	0.65	0.68	0.65	0.50	0.50
Knowledge	0.18	0.16	0.22	0.18	0.22	0.18	0.22	0.45	0.50

Similarity = 0.25

Parameter	Shock	Turnover		Rotation		TMT (0.01)		TMT (0.1)	
		Low	*High*	*Low*	*High*	*Low*	*High*	*Low*	*High*
Unlocking	0.33	0.27	0.45	0.59	0.80	0.58	0.65	0.99	0.99
Diversity	0.94	0.96	0.95	0.92	0.90	0.92	0.90	0.53	0.51
Knowledge	0.08	0.08	0.14	0.15	0.19	0.17	0.19	0.70	0.73

Finally, the findings also have practical implications for organizations and their management. First, the management needs to observe the diversity in the organization during the path formation process in order to decide which mean is appropriate to unlock paths (e.g. *turnover* or *rotation*). Measuring heterogeneity is a difficult endeavor, but adequate methods are already available. For example, measuring the degree of heterogeneity through a real options analysis is still a little used, but powerful, method to evaluate the scope of strategic choice (McGrath, et al., 2004). If managers notice a decline in the number of real options, they might take action by creating organizational niches with diverse individuals (Hannan, et al., 2003). The results of the simulation experiments show that organizational niches with distinct groups are a ‘breeding ground’ for diverse beliefs and can help to unlock paths. Second, in the case of perceived or real strategic inflexibility, the suggested means may give guidance to managers or the board of directors. If an internal analysis shows that the organization does not possess heterogeneity, such as in the form of slack resources and real options, integrating external knowledge through turnover is advisable. On the other hand, if the organizations can draw on a diverse set of beliefs, it should aim for reconfigurations and coordinate through the implementation of a new strategy. However, resistance to change can be a barrier to intentional actions through the management or board and must be taken into account (see for example Ford, et al. (2008)). Third, to prevent a detrimental lock-in situation, managers should take preventive measures through supporting heterogeneity within an organization. In the R&D and innovation management literature several practices to enhance heterogeneity are known. Without any claim for completeness, activities like corporate venturing (Block & MacMillan, 1995), open innovation (Chesbrough, 2003), crowdsourcing (Affuah & Tucci, 2012), skunk works (Fosfuri & Ronde, 2009), business accelerators (Wolcott & Lippitz, 2007), and flexible R&D management (Niosi, 1999) may potentially increase heterogeneity.

The next chapter concludes the dissertation by giving a brief summary and hinting to limitations of the work and future research.

7. Summary, Limitations & Further Research

7.1 Summary

The present work is an early attempt to integrate unlocking into the concept of organizational path dependence. Building upon prior management literature on path dependence, necessary conditions and means to unlock paths were derived in order to come up with a simulation model. The simulation model took into account that individuals in organizations are subject to restrictions when learning from other agents. Following a simulation research protocol, the model was translated into computer code to perform virtual experiments. The baseline model proved that under the constituting properties of a path dependent process, self-reinforcing mechanisms will lead to organizational lock-in. Furthermore, it has been shown that isolated groups can emerge through the similarity based selection and learning process, even under the properties of a path-dependent process. The more groups emerge, the higher the diversity of beliefs in the organization. In the case of a static environment, belief diversity impedes proper adaptation, and therefore results in poor firm performance, represented through the average individual knowledge within an organization. These findings highlight the importance of the similarity based individual selection and learning mechanism in organizations, and extend prior research on organizational learning and path dependence. In accordance with the definition of path dependence, diversity alone does not allow for unlocking in a static environment. The diversity captures the variation present in social behavior, and therefore reflects the notion of the "shadow" in the lock-in phase of the three-phase model of path dependence. Even when diversity is present within an organization, path dependence can emerge on organizational level and prevent adaptation towards an environment in the lock-in phase. This finding is important, because without diversity, unlocking of paths is not possible. With the base model reflecting a path dependent process, a trigger for unlocking needs to be induced. Hence, the first set of experiments integrated an exogenous shock into the simulation model, and demonstrated that through changes in the environment, organizational transformation processes may be triggered. The results show that if an exogenous shock occurs, unlocking is only possible for environments that share similarities with the prior environment, or for organizations consisting of different groups holding diverse set of beliefs. But the results also emphasize that too much diversity, emerging from high similarity values, hampers effective adaptation to a changed environment. Under the assumption of the selection and learning mechanism, these findings suggest that organizations should not strive for too much diversity in order to unlock paths, but instead find a balance between diversity and adaptation towards the environment. With these results at hand, it can be stated that the presence of an exogenous shock and heterogeneity is important for endogenously unlocking paths. As the environment changes, and therefore also the selection criteria, agents may switch their learning partner and the recombination of beliefs leads through a chain reaction to adaptation to the new environment. Hence, different to prior simulations in path research, the study removes limitations by taking into account the organizational structure, and shows that structural changes, initiated by the same learning mechanism that lead to path dependence, can also lead to the unlocking of paths. With regard to the first research question, it can hence be said that the results of the simulation confirm that paths can be endogenously unlocked in the presence of an exogenous shock and heterogeneity, according to the four-phase path model proposed by Sydow, et al. (2005).

After it has been proven that the dissolution of paths is indeed possible, even without inducing new beliefs from outside of the organization, the second set of experiments tested how paths can be unlocked through intentional actions. With turnover and rotation, two distinct approaches, proposed by the literature as means to break paths, have been tested. Turnover induces new beliefs from outside into the organization, while rotation makes use of already existing beliefs. The results show that for low similarity values, high turnover proves to be very effective to counteract the detrimental effects of an environmental shock, while for high similarity values reconfiguring the organization through rotation increases the probability of path unlocking. Furthermore, through intentional means the probability of unlocking increases, compared to endogenous path dissolution.

Because prior research has shown that a hierarchy influences the path formation process, it may be assumed that it also affects the unlocking of paths. Therefore, in the third set of experiments, a hierarchical level, represented by a dominant top management team, was added to the simulation model to investigate the influence of a hierarchy on the ability to unlock paths. The findings imply that, while authorities impose their beliefs on individuals, the resulting coordination effects lead to a higher average individual knowledge in the path formation process for similarity values greater than 0.1, but at the same time hamper the probability of unlocking. Exchanging members of the top management team after a shock occurred is, like turnover, inducing belief variety in the organization and proves to be a powerful mean to unlock paths. With respect to the second research question, it can therefore be stated that rotation, turnover, and top management team influence have a positive impact on the probability of unlocking. Without taking similarity into account, labor and top management team turnover prove to be most effective, while for medium and high similarity values rotation and coordination through a top management team are more effective.

While the findings extent the theory of path dependence, by showing how unlocking can occur, special caution must be exercised, because of the limitations of the study.

7.2 Limitations

While taking great care to ensure accuracy, the conclusions drawn from the findings of the simulation study also exhibit limitations. Like every other method, computer simulations in general have a number of disadvantages. First, as computer simulations are based on virtually constructed worlds, no statements can be made if the findings hold in the real world. Therefore, it is proposed to examine the findings empirically. As pointed out in Chapter 3, laboratory experiments are a good mean to prove path formation empirically, and could also help in identifying the unlocking of paths. With the help of experiments, the findings should be reproduced and independently verified in a social context. Again, a precautionary approach has to be taken in order to model path dependence accurately. As a simulation model is always an abstraction from reality, and is not able to capture all possible interdependencies, omitted moderating variables could influence the results. For example, dissimilarity in organizations is here treated as a mean to induce heterogeneity. But, in reality, heterogeneity also negatively influences turnover rates (Jackson, et al., 1991). These interdependencies are not considered in the current model. But, by replicating and building upon a proven organizational learning model, it is expected that findings will correspond with the reality. Also, while the code was debugged, the probability of errors cannot be completely ruled out. However, using the proposed four-phase approach to detect inconsistencies, major errors should be avoided. Further limitations of computer simulations in the field of path research can be found in Petermann (2010) and Seidel (2013). Furthermore the similarity-based selection and learning algorithm must be investigated to make sure that the behavior occurs in organizations and has the proposed outcomes on unlocking. The segregation effect of the learning mechanism needs therefore to be elaborated empirically. For example, a study about learning in different organizational units after restructuring could generate new insights on how learning takes place in organizations between dissimilar individuals.

7.3 Further Research

While this work provides a first initial step on the topic of unlocking, more research must be conducted to fully understand the concept. But, besides conducting empirical research, the simulation model itself opens up several possibilities for further extensions. Unlocking in the present model is possible because of the heterogeneity, arising of the similarity-based selection and learning algorithm. Beyond that, further approaches to unlock paths, such as errors appearing during social interactions (K. D. Miller, et al., 2006), variation mechanisms in the learning and communication process (Pentland, et al., 2012), or imperfect social adaptation (Walsh & Ungson, 1991), could be used to show how unlocking can occur. The micro-foundations of organizations may hence help to better understand the concept of unlocking. Also, while it is argued that the simple selection and learning rule captures the social behavior of individuals in the organization, a more complex rule might depict organizational behavior more accurately and reveal new insights. For example, agents could base their decision on who to interact with more advanced criteria, like differently-weighted similarity categories or other barriers of tapping into new knowledge. A good starting point is the innovation management literature and research on innovation barriers. One can also think of different rule guided individual behavior besides the applied similarity rule such as conformity, opportunism, or creativity, and examine how these micro behaviors alter the outcome of the simulation. Furthermore, the model assumes that each agent is linked to four of its neighbors. The results may change if the number of links or the distance between agents is increased, making it possible for agents to tap into knowledge of distant agents. Furthermore, more complex organizational structures could be depicted using the existing model. One could drill deeper into the group structure, and have a look on how group size, shape of groups, or distance of individual belief sets amount to the probability of unlocking paths. Expected findings could hint to organizational structures that are more resilient for unlocking organizational paths, and may guide organizational designers. As individuals in organizations are seldom the ones determining the structure through their emergent behavior, the results might change if the structure is kept fixed or initially specified by the management. This work shows that heterogeneity plays a major role in unlocking of organizational paths. But while the model at hand is a first avenue to highlight heterogeneity, it does not go into the concept in depth. Future work should hence dig deeper by breaking down heterogeneity into different items. For example, the distance of belief sets between different groups in the organization, the number of different belief sets, or how disperse the set of beliefs are distributed within the organization could be measured. Some of the measurement parameters are already included in the computer code, and mechanisms for the evaluation have been defined. The model can be extended to accommodate some of the aforementioned concepts. While there could possibly be more limitations, at last some concluding remarks will be mentioned.

7.4 Concluding Remarks

The results of the simulation model confirm that unlocking in organizations can occur and show means for intentionally unlocking paths. Although simulation studies simplify social processes, the generated findings provide a guideline for empirical research. Research on technological path dependence started with a formal model and was then successively tested empirically. Therefore, starting off with a simulation model could stimulate the discussion on unlocking, and give an impulse to further empirical research. The call for more research on unlocking of organizational paths, in order to give a more complete understanding of path dependence, is shared by researchers. Now that unlocking according to the four-phase path model is confirmed, and potential drivers for unlocking are identified, the construct can be explored deeper to extend the theory of path *in*dependence. The rich knowledge about how organizations could unlock paths may also inform practitioners on how to design organizations and react to lock-in. A better understanding about the connection between heterogeneity, lock-in and unlocking can be a guideline for designing organizations and initiating change process. Remarkable, heterogeneity does not automatically imply flexibility, but is moderated by the organizational structure and learning processes. Therefore, only inducing heterogeneity is not sufficient to unlock paths. In summary, the dissertation shows the importance of similarity, heterogeneity, learning, structure, and intentional means to unlock paths and serves as a first reference and stimulus for further research in this direction. Picking up the example of IBM from the introduction of the dissertation, and taking these results as a basis, one could assume that IBM was able to change because of heterogeneous resources within the organization. Through structural integration of heterogeneous beliefs, and coordination through a new top management as an intentional mean to unlock paths, IBM could have been able to adapt to a changed environment. Effectively, following the story of Louis Gerstner, it was not only him as a former CEO driving the change of IBM into a modern service-oriented company, but also the slack resources available within the organization. Yet, if a similarity based learning mechanism lead to the evolution of such slack resources, remains unexplored.

References

Affuah, A., & Tucci, C. L. (2012). Crowdsourcing as a Solution to Distant Search. *Academy of Management Review, 37*, 355-375.

Akerlof, G. A., & Dickens, W. T. (1983). The Economic Consequences of Cognitive Dissonance. *American Economic Review, 72*, 307-319.

Alexander, J., & Giesen, B. (1987). From Reduction to Linkage: The Long View of the Micro-Macro Link. In J. Alexander, B. Giesen, R. Muench & N. J. Smelser (Eds.), *The Micro-Macro Link* (pp. 45-71). Berkely: University of California Press.

Alexander, J., Nuchols, B., Bloom, J., & Lee, S. (1995). Organizational Demography and Turnover: An Examination of Multiform and Nonlinear Heterogeneity. *Human Relations, 48*, 1455-1480.

Allen, P. M. (1988). Evolution, Innovation and Economics. In G. Dosi, C. Freeman, R. Nelson, G. Silverberg & L. Soete (Eds.), *Technical Change and Economic Theory* (pp. 95-119). London: Pinter.

Almeida, C. M., Gleriani, J. M., Castejon, E. F., & Soares-Filho, B. S. (2008). Using Neural Networks and Cellular Automata for Modelling Intra‐Urban Land‐Use Dynamics. *International Journal of Geographical Information Science, 22*, 943-963.

Anderlini, L., & Ianni, A. (1996). Path Dependence and Learning from Neighbors. *Games and Economic Behavior, 13*, 141-177.

Araujo, L., & Harrison, D. (2002). Path Dependence, Agency and Technological Evolution. *Technology Analysis & Strategic Management, 14*, 5-19.

Argote, L. (1999). *Organizational Learning: Creating, Retaining, and Transferring Knowledge*. Boston: Kluwer Academic.

Argote, L., Beckman, S. L., & Epple, D. (1990). The Persistence and Transfer of Learning in Industrial Settings. *Management Science, 36*, 140-154.

Argyris, C., & Schoen, D. A. (1978). *Organizational Learning: A Theory of Action Perspective*. Reading, MA: Addison-Wesley

Arrow, H., McGrath, J. E., & Berdahl, J. L. (2000). *Small Groups as Complex Systems: Formation, Coordination, Development, and Adaptation*. London: Sage.

Arrow, K. J. (1962). The Economic Implications of Learning by Doing. *The Review of Economic Studies, 29*, 155-173.

Arthur, W. B. (1983). Competing Technologies and Lock-In by Historical Small Events: The Dynamics of Allocation under Increasing Returns. In: Stanford University.

Arthur, W. B. (1989). Competing Technologies, Increasing Returns, and Lock-In by Historical Events. *Economic Journal, 99*, 116-131.

Arthur, W. B. (1994). *Increasing Returns and Path Dependence in the Economy*. Ann Arbor: University of Michigan Press.

Arthur, W. B. (2009). *The Nature of Technology: What it is and how it Evolves*. New York: The Free Press

Ashby, W. R. (1956). *An Introduction to Cybernetics*. London: Chapman and Hall.

Axelrod, R. (1997a). Advancing the Art of Simulation in the Social Science. In R. Conte, R. Hegselmann & P. Terna (Eds.), *Simulating Social Phenomena* (pp. 21-40). Berlin: Springer-Verlag.

Axelrod, R. (1997b). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princetion: Princeton University Press.

Axelrod, R. (2003). Adavancing the Art of Simulation in the Social Science. *Journal of the Japan Society for Management Information, 12*, 3-16.

Axelrod, R. (2007). Simulation in the Social Science. In J. Rennard (Ed.), *Handbook of Research on Nature Inspired Computing for Economics and Management* (Vol. 1, pp. 90-100). Hershey: Idea Group Inc.

Axtell, R., Axelrod, R., Epstein, J. M., & Cohen, M. D. (1996). Aligning Simulation Models: A Case Study and Results. *Computational & Mathematical Organization Theory, 1*, 123-141.

Balmann, A. (1994). *Pfadabhängigkeiten in Agrarstrukturentwicklungen - Begriff, Ursachen und Konsequenzen.* Universität Göttingen, Göttingen.

Barker III, V. L., Patterson Jr., P. W., & Mueller, G. C. (2001). Organizational Causes and Strategic Consequences of the Extent of Top Management Team

Replacement During Turnaround Attempts. *Journal of Management Studies, 38*, 235-270.

Barney, J. B. (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management, 17*, 99-120.

Barney, J. B. (2001). Resource-Based Theories of Competitive Advantage: A Ten-Year Retrospective on the Resource-Based View. *Journal of Management, 27*, 643-650.

Bassanini, A. P., & Dosi, G. (2000). Heterogeneous Agents, Complementarities, and Diffusion: Do Increasing Returns Imply Convergence to International Technological Monopolies? In D. D. Gatti, M. Gallegati & A. Kirman (Eds.), *Market Structure, Aggregation and Heterogeneity*. Berlin: Springer.

Bassanini, A. P., & Giovanni, D. (2001). When and How Chance and Human Will Can Twist the Arms of Clio: An Essay on Path Dependence in a World of Irreversibilities. In R. Garud & P. Karnoe (Eds.), *Path Dependence and Creation* (pp. 41-69). Mahwah: Lawrence Erlbaum Associates.

Baumann, O. (2008). *Problem Solving in Complex Systems: Essays on Search, Design, and Strategy.* Ludwig-Maximilans-University Munich, Munich.

Becker, M. C. (2006). The Concept of Routines Twenty Years after Nelson and Winter (1982): A Review of the Literature In *Danish Research Unit for Industrial Dynamics*. Odense: University of Southern Denmark.

Beckmann, C. M., & Burton, M. D. (2008). Founding the Future: Path Dependence in the Evolution of Top Management Teams from Founding to IPO. *Organization Science, 19*, 3-24.

Beltrami, E. J. (1999). *What is Random? - Chance and Order in Mathematics and Life*. New York: Springer-Verlag.

Benner, M. J., & Tushman, M. L. (2002). Process Management and Technological Innovation: A Longitudinal Study of the Photography and Paint Industries. *Administrative Science Quarterly, 47*, 676-706.

Beyer, J. (2010). The Same or Not the Same - On the Variety of Mechanisms of Path Dependence. *International Journal of Human and Social Sciences, 5*, 1-10.

Biggart, N. W. (1977). The Creative-Destructive Process of Organizational Change: The Case of the Post Office. *Administrative Science Quarterly, 22*, 410-426.

Billari, F. C., Fent, T., Prskawetz, A., & Scheffran, J. (2006). Agent-Based Computational Modelling: An Introduction. In F. C. Billari, T. Fent, A. Prskawetz & J. Scheffran (Eds.), *Agent-Based Computational Modelling: Applications in Demography, Social, Economic and Environment Sciences* (pp. 1-16). Heidelberg: Physica-Verlag HD.

Bingham, C. B., Eisenhardt, K. M., & Furr, N. R. (2007). Microfoundations of Performance: Balancing Efficiency and Flexibility in Dynamic Environments. *Organization Science, 21*, 1263-1273.

Birkhoff, G. D. (1931). Proof of the Ergodic Theorem. *Proceedings of the National Academy of Sciences of the United States of America, 17*, 656.

Block, Z., & MacMillan, I. C. (1995). *Corporate Venturing: Creating New Businesses within the Firm*. Cambridge, MA.: Harvard Business School Press.

Bonaccorsi, A., & Rossi, C. (2003). Why Open Source Software can Succeed. *Research Policy, 32*, 1243-1258.

Bourgeois III, L. J. (1981). On the Measurement of Organizational Slack. *Academy of Management Review, 6*, 29-39.

Bourgeois III, L. J. (1984). Strategic Management and Determinism. *Academy of Management Review, 9*, 586-596.

Bowman, E. H., & Singh, H. (2007). Corporate Restructuring: Reconfiguring the Firm. *Strategic Management Journal, 14*, 5-14.

Boyes, W., & Melvin, M. (2008). *Microeconomics*. Boston, MA: Houghton Mifflin Company.

Bramson, A. (2008). Measures of Tipping Points, Robustness, and Path Dependence. In (pp. 1-51): Cornell University Library.

Brandte, H. (2007). *Komplexität in Organisationen - Organisationstheoretische Betrachtungen und agentenbasierte Simulation*. Wiesbaden: Deutscher Universitäts-Verlag.

Brown, D. G., Page, S., Riolo, R., Zellner, M., & Rand, W. (2005). Path Dependence and the Vaildation of Agent-Based Spatial Models of Land Use. *International Journal of Geographical Information Science, 19*, 153-174.

Burgelman, R. A. (1983). A Process Model of Internal Corporate Venturing in the Diversified Major Firm. *Administrative Science Quarterly, 28*, 223-244.

Burgelman, R. A. (1991). Intraorganizational Ecology of Strategy Making and Organizational Adaptation: Theory and Field Research. *Organization Science, 2*, 239-262.

Burgelman, R. A. (1994). Fading Memories: A Process Theory of Strategic Business Exit in Dynamic Environments. *Administrative Science Quarterly, 39*, 24-56.

Burgelman, R. A., & Grove, A. S. (1996). Strategic Dissonance. *California Management Review, 38*, 8-25.

Burns, T., & Stalker, G. M. (1961). *The Management of Innovation*. London: Travistock.

Byrne, D. (1971). *The Attraction Paradigm*. New York: Academic Press.

Byrne, D., & Nelson, D. (1964). Attraction as a Function of Attitude Similarity–Dissimilarity: The Effect of Topic Importance. *Psychonomic Science, 1*, 93-94.

Cabrera, A., & Cabrera, E. F. (2002). Knowledge-Sharing Dilemmas. *Organization Studies, 23*, 687-710.

Camerer, C. (1998). Bounded Rationality in Individual Decision Making. *Experimental Economics, 1*, 163-183.

Campion, M., Cheraskin, L., & Stevens, M. J. (1994). Career-Related Antecedents and Outcomes of Job Rotation. *Academy of Management Journal, 37*, 1518-1542.

Carley, K. M. (1995). Computational and Mathematical Organization Theory: Perspective and Directions. *Mathematical Organization Theory, 1*, 39-56.

Carley, K. M., & Newell, A. (1994). The Nature of Social Agent. *Journal of Mathematical Sociology, 19*, 221-262.

Carroll, G. R., & Harrison, R. J. (1994). On the Historical Efficiency of Competition between Organizational Populations. *American Journal of Sociology, 100*, 720-749.

Castaldi, C., & Dosi, G. (2005). *The Grip of History and the Scope of Novelty. Some Results and Open Questions on Path Dependence*. Basingstoke: Palgrave Macmillan.

Casti, J. L. (1996). *Would-be Worlds: How Simulation is Changing the Frontiers of Science*. New York: John Wiley & Sons.

Chattoe, E., Saam, N. J., & Moehring, M. (2000). *Sensitivity Analysis in the Social Sciences: Problems and Prospects*. Heidelberg: Physica-Verlag

Cheng, J. L., & Kesner, I. F. (1997). Organizational Slack and Response to Environmental Shifts: The Impact of Resource Allocation Patterns. *Journal of Management, 23*, 1-18.

Chesbrough, H. W. (2003). *Open innovation: The New Imperative for Creating and Profiting from Technology*: Harvard Business Press.

Child, J. (1972). Organizational Structure, Environment and Performance: The Role of Strategic Choice. *Socioloy, 6*, 1-22.

Christensen, C. M. (1997). *The Innovator's Dillemma: When New Technologies Cause Great Firms to Fail*. Cambridge, MA: Harvard Press.

Christensen, C. M., & Rosenbloom, R. S. (1995). Explaining the Attacker's Advantage: Technological Paradigms, Organizational Dynamics, and the Value Network. *Research Policy, 24*, 233-257.

Cohen, M. D., & Bacdayan, P. (1994). Organizational Routines are Stored as Procedural Memory: Evidence from a Laboratory Study. *Organization Science, 5*, 554-568.

Cohen, M. D., March, J. G., & Olsen, J. P. (1972). A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly, 17*, 1-25.

Cohen, W. D., & Levinthal, D. A. (1990). Absorptive Capacity: A New Perspective on Learning and Innovation. *Administrative Science Quarterly, 35*, 128-152.

Conte, R., Edmonds, B., Moss, S., & Sawyer, R. K. (2001). Sociology and Social Theory in Agent Based Social Simulation: A Symposium. *Computational & Mathematical Organization Theory, 7*, 183-205.

Coriat, B., & Dosi, G. (1998). The Institutional Embeddedness of Economic Change. An Appraisal of the "Evolutionary" and the "Regulationist" Research Programs. In K. Nielsen & B. Johnson (Eds.), *Institutions and Economic Change*. Cheltenham, UK: Edward Elgar.

Cowan, R. (1990). Nuclear Power Reactors: A Study in Technological Lock-In. *The Journal of Economic History, 50*, 541-567.

Cowan, R., & Gunby, P. (1996). Sprayed to Death: Path Dependence, Lock-In and Pest Control Strategies. *The Economic Journal, 106*, 521-542.

Cowan, R., & Hulten, S. (1996). Escaping Lock-In: The Case of the Electric Vehicle. *Technological Forecasting and Social Change, 53*, 61-79.

Crossan, M. M., Lane, H. W., & White, R. E. (1999). An Organizational Learning Framework: From Intuition to Institution. *The Academy of Management Review, 24*, 522-537.

Cusumano, M. A., Mylonadis, Y., & Rosenbloom, R. S. (1992). Strategic Maneuvering and Mass-Market Dynamics: The Triumph of VHS over Beta. *Business History Review, 66*, 51-94.

Cyert, R. M., & March, J. G. (1963). *A Behavioral Theory of the Firm*. Englewood Cliffs, NJ: Prentice Hall.

Darr, E. D., & Kurtzberg, T. (2000). An Investigation of Partner Similarity Dimensions on Knowledge Transfer. *Organizational Behavior and Human Decision Processes, 82*, 28-44.

Das, M. (2006). Formal Specifications on Industrial-Strength Code—From Myth to Reality. In T. Ball & R. Jones (Eds.), *Computer Aided Verification* (pp. 1). Berlin: Springer.

DasGupta, A. (2008). *Asymptotic Theory of Statistics and Probability*. Berlin: Springer.

David, P. A. (1985). Clio and the Economics of QWERTY. *American Economic Review, 75*, 332-337.

David, P. A. (1997). Path Dependence and the Quest for Historical Economics: One More Chorus of the Ballad of QWERTY. In *Discussion Papers in Economic and Social History*: University of Oxford.

David, P. A. (1999). At Last, a Remedy for Chronic QWERTY-Skepticism! In *Discussion Paper for the European Summer School in Industrial Dynamics*.

David, P. A. (2001). Path Dependence, its Critics and the Quest for 'Historical Economics'. *Evolution and path dependence in economic ideas: Past and present, 15*, 40.

David, P. A. (2007). Path Dependence: A Foundational Concept for Historical Social Science. *Cliometrica, 1*, 91-114.

David, P. A., & Bunn, J. A. (1988). The Economics of Gateway Technologies and Network Evolution: Lessons from Electricity Supply History. *Information Economics and Policy, 3*, 165-202.

Davis, J. P., Eisenhardt, K. M., & Bingham, C. B. (2007). Developing Theory through Simulation Methods. *Academy of Management Review, 32*, 480-499.

Dawkins, R. (1996). *The Blind Watchmaker*. New York: W. W. Norton & Company.

de Rond, M., & Thietart, R. (2007). Choice, Chance, and Inevitability in Strategy. *Strategic Management Journal, 28*, 535-551.

del Rio, P., & Unruh, G. (2007). Overcoming the Lock-Out of Renewable Energy Technologies in Spain: The Cases of Wind and Solar Electricity. *Renewable and Sustainable Energy Reviews, 11*, 1498-1513.

Dixit, A. K., & Stiglitz, J. E. (1977). Monopolistic Competition and Optimum Product Diversity. *The American Economic Review, 67*, 297-308.

Dobusch, L., & Kapeller, J. (2012). Striking New Paths: Theory and Method in Path Dependence Research. In. Berlin: Freie Universität Berlin.

Dodgson, M. (1993). Organizational Learning: A Review of Some Literatures. *Organization Studies, 14*, 375-394.

Dolfsma, W., & Leydesdorff, L. (2009). Lock-In and Break-Out from Technological Trajectories: Modelling and Policy Implications. *Technological Forecasting and Social Change, 76*, 932-941.

Dosi, G., Marengo, L., Paraskevopoulou, E., & Valente, M. (2011). The Value and Dangers of Rememberance in Changing Worlds: A Model of Cognitive and Operational Memory of Organizations. In *DIME Final Conference*. Maastricht.

Dougherty, D. (1992). Interpretive Barriers to Successful Product Innovation in Large Firms. *Organization Science, 3*, 179-202.

Duncan, R. (1972). Characteristics of Organizational Environments and Perceived Environmental Uncertainty. *Administrative Science Quarterly, 17*, 313-327.

Eldredge, N., & Gould, S. J. (1972). Punctuated Equilibria: An Alternative to Phyletic Gradualism. In T. J. Schop (Ed.), *Models in Paleobiology* (pp. 82-115). San Francisco: Freeman, Cooper & Co.

Ellis, D. G. (1999). Research on Social Interaction and the Micro-Macro Issue. *Research on Language and Social Interaction, 32*, 31-40.

Ericson, M., & Lundin, R. A. (2013). Locking In and Unlocking - Adding to Path Dependence. In J. Sydow & G. Schreyögg (Eds.), *Self-Reinforcing Processes in and among Organizations* (pp. 185-204). New York: Palgrave Macmillan.

Ethiraj, S. K., & Levinthal, D. A. (2004). Modularity and Innovation in Complex Systems. *Management Science, 50*, 159-173.

Ethiraj, S. K., & Levinthal, D. A. (2009). Complex Organizations and Multiple Goals. *Organization Science, 20*, 4-21.

Ethiraj, S. K., Levinthal, D. A., & Roy, R. R. (2008). The Dual Role of Modularity: Innovation and Imitation. *Management Science, 54*, 939-955.

Fang, C., Lee, J., & Schilling, M. A. (2010). Balancing Exploration and Exploitation through Structural Design: The Isolation of Subgroups and Organizational Learning. *Organization Science, 21*, 625-642.

Farrell, J., & Saloner, G. (1986). Installed Base and Compatibility: Predation, Product Preannouncements and Innovation. *American Economic Review, 76*, 940-955.

Feinstein, A. H., & Cannon, H. M. (2002). Constructs of Simulation Evaluation. *Simulation Gaming, 33*, 425-440.

Festinger, L. (1950). Informal Social Communication. *Psychological Review, 57*, 271-282.

Festinger, L. (1957). *A Theory of Cognitive Dissonance*. New York: Row, Peterson and Company.

Festinger, L. (1964). *Conflict, Decision, and Dissonance*. Stanford, CA: Stanford University Press.

Fiol, M. C. (1994). Consensus, Diversity, and Learning in Organizations. *Organization Science, 5*, 403-420.

Ford, J., Ford, L., & D'Amelio, D. (2008). Resistance to Change: The Rest of the Story. *Academy of Management Review, 33*, 362-377.

Forrester, J. W. (1958). Industrial Dynamics: A Major Breakthrough for Decision Makers. *Harvard Business Review, 36*, 37-66.

Forrester, J. W. (1980). *Principles of Systems*. Cambridge, MA: MIT Press.

Fosfuri, A., & Ronde, T. (2009). Leveraging Resistance to Change and the Skunk Works Innovation. *Journal of Economic Behaviour & Organization, 72*, 274-289.

Freedman, J. (1965). Preference for Dissonant Information. *Journal of Personality and Social Psychology, 2*, 195-202.

Frenken, K. (2006). *Innovation, Evolution and Complexity Theory*. Cheltenham, UK: Edward Elgar Publishing Limited.

Friedkin, N. E., & Marsden, P. (1994). *Network Studies of Social Influence*. Thousand Oaks, CA: Sage.

Friedman, D. D. (1986). *Price Theory: An Intermediate Text*. Cincinnati, OH: South-Western Publishing Co.

Friedman, R. A., & Podolny, J. (1992). Differentiation of Boundary Spanning Roles: Labor Negotiations and Implications for Role Conflict. *Administrative Science Quarterly, 37*, 28-47.

Galbraith, J. R. (1982). Designing the Innovating Organization. *Organizational Dynamics, 10*, 5-25.

Ganco, M., & Hoetker, G. (2009). NK Modelling Methodology in the Strategy Literature: Bounded Search on a Rugged Landscape. In D. D. Bergh & D. J. Ketchen (Eds.), *Research Methodoloy in Strategy and Management*. Bingley, UK: JAI Press.

Gandal, N., Kende, M., & Rob, R. (2000). The Dynamics of Technological Adoption in Hardware/Software Systems: The Case of Compact Disc Players. *The RAND Journal of Economics, 31*, 43-61.

Garud, R., & Karnoe, P. (2001). Path Creation as a Process of Mindful Deviation. In R. Garud & P. Karnoe (Eds.), *Path Dependence and Creation* (pp. 1-38). New York: Lawrence Earlbaum Assoicates.

Garud, R., & Karnoe, P. (2003). Bricolage versus Breakthrough: Distributed and Embedded Agency in Technology Entrepreneurship. *Research Policy, 32*, 277-300.

Garud, R., Karnoe, P., & Kumaraswamy, A. (2010). Path Dependence and Creation. *Journal of Management Studies, 47*, 760-774.

Gavetti, G. (2005). Cognition and Hierarchy: Rethinking the Microfoundations of Capabilities Development. *Organization Science, 16*, 599-617.

Gavetti, G., Greve, H. R., Levinthal, D. A., & Ocasio, W. (2012). The Behavioral Theory of the Firm: Assessment and Prospects. *The Academy of Management Annals, 6*, 1-40.

Gavetti, G., & Levinthal, D. A. (2000). Looking Forward and Backward: Cognitive and Experiential Search. *Administrative Science Quarterly, 45*, 113-137.

Gavetti, G., Levinthal, D. A., & Rivkin, J. W. (2005). Strategy Making in Novel and Complex Worlds: The Power of Analogy. *Strategic Management Journal, 26*, 691-712.

Gersick, C. J. G. (1991). Revolutionary Change Theories: A Multilevel Exploration of the Punctuated Equilibrium Paradigm. *Academy of Management Review, 16*, 10-36.

Gerstner, L. V. (2002). *Who says Elephants can't Dance? Inside IBM's Historic Turnaround*. New York, NY: HarperBusiness.

Ghemawat, P. (1991). *Flexibility and Commitment: The Dynamics of Strategy*. New York: Free Press.

Gilbert, C. G. (2005). Unbundling the Structure of Inertia: Resource versus Routine Rigidity. *Academy of Management Journal, 48*, 741-763.

Gilbert, N. (1995). *Emergence in Social Simulation*. London: UCL Press.

Gilbert, N., & Terna, P. (2000). How to Build and use Agent-Based Models in Social Science. *Mind & Society, 1*, 57-72.

Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the Social Scientist*. New York: Open University Press.

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik, 38*, 173-198.

Goldspink, C., & Kay, R. (2004). Bridging the Micro-Macro Divide: A New Basis for Social Science. *Human Relations, 57*, 597-618.

Gould, S. J. (1977). *Ever since Darwin*. New York: Norton.

Gould, S. J. (1980). Is a New and General Theory of Evolution Emerging? *Paleobiology, 6*, 119-130.

Gould, S. J. (2002). *The Structure of Evolutionary Theory*. Cambridge, MA: Harvard University Press.

Gould, S. J. (2007). *Punctuated Equilibrium*. Cambridge, MA: Belknap Press of Harvard University Press.

Greve, H. R. (1998). Performance, Aspirations, and Risky Organizational Change. *Administrative Science Quarterly, 43*, 58-86.

Greve, H. R. (2007). Exploration and Exploitation in Product Innovation. *Industrial and Corporate Change, 16*, 945-975.

Gupta, A. K., Smith, K. G., & Shalley, C. E. (2008). The Interplay between Exploration and Exploitation. *Academy of Management Journal, 49*, 693-706.

Hambrick, D. C., Seung Cho, T., & Chen, M. (1996). The Influence of Top Management Team Heterogeneity on Firms' Competitive Moves. *Administrative Science Quarterly, 41*, 659-684.

Hand, C. (2006). History Matters: Modelling Path Dependence on a Spreadsheet. *Computers in Higher Education Economics Review, 18*, 19-24.

Hannan, M. T., Carroll, G. R., & Polos, L. (2003). The Organizational Niche. *Sociological Theory, 21*, 309-340.

Hannan, M. T., & Freedman, J. (1977). The Population Ecology of Organizations. *American Journal of Sociology, 82*, 929-964.

Hannan, M. T., & Freeman, J. (1984). Structural Inertia and Organizational Change. *American Sociological Review, 49*, 149-164.

Harrison, D. A., & Klein, K. J. (2007). What's the Difference? Diversity Constructs as Separation, Variety or Disparity in Organizations. *Academy of Management Review, 32*, 1199-1228.

Harrison, J. R. (2008). Dominant Coalition Dynamics: The Politics of Organizational Adaptation and Failure. In *The Fifth Conference of The European Social Simulation Association*. Brescia, IT.

Harrison, J. R., Carroll, G. R., & Carley, K. M. (2007). Simulation Modelling in Organizational and Management Science. *Academy of Management Review, 32*, 1229-1245.

Hassink, R. (2005). How to Unlock Regional Economies from Path Dependency? From Learning Region to Learning Cluster. *European Planning Studies, 13*, 521-535.

He, Z., & Wong, P. (2004). Exploration vs. Exploitation: An Empirical Test of the Ambidexterity Hypothesis. *Organization Science, 15*, 481-494.

Hegselmann, R. (1996). Cellular Automata in the Social Sciences: Prespectives, Restrictions and Artefacts. In R. Hegselmann, N. Gilbert, U. Mueller, J. E. Doran & K. G. Troitzsch (Eds.), *Social Science Microsimulation* (pp. 282-306). Berlin: Springer-Verlag.

Henderson, R. (1993). Underinvestment and Incompetence as Responses to Radical Innovation: Evidence from the Photolithographic Alignment Equipment Industry. *The RAND Journal of Economics, 24*, 248-270.

Hendron, M. G., Bednar, M. K., & Henderson, A. D. (2005). How Aspiration Levels Affect Response to a Punctuating Technological Change: An Empirical Test. *Academy of Management Proceedings, August 2005*, 1-6.

Hill, C. W. (1997). Establishing a Standard: Competitive Strategy and Technological Standards in Winner-Take-All Industries. *The Academy of Management Executive, 11*, 7-25.

Hite, J., & Hesterly, W. (2001). The Evolution of Firm Networks: From Emergence to Early Growth of the Firm. *Strategic Management Journal, 22*, 275-286.

Hodgson, G. M., & Knudsen, T. (2006). Balancing Inertia, Innovation, and Imitation in Complex Environments. *Journal of Economic Issues, 40*, 287-295.

Hollenbeck, J. R., Ilgen, D. R., Tuttle, D., & Sego, D. J. (1995). Team Performance on Monitoring Tasks: An Examination of Decision Errors in Contexts Requiring Sustained Attention. *Journal of Applied Psychology, 80*, 685-696.

Holtmann, J. P. (2008). *Pfadabhängigkeit strategischer Entscheidungen: Eine Fallstudie am Beispiel des Bertelsmann Buchclubs Deutschland*. Wiesbaden: Gabler Verlag.

Horner, J. (1991). The Case of DAT Technology: Industrial versus Pecuniary Function. *Journal of Economic Issues, 25*, 449-457.

Inkpen, A., & Choudhury, N. (1995). The Seeking of Strategy where it is not: Towards a Theory of Strategy Absence. *Strategic Management Journal, 16*, 313-323.

Jackson, S., Brett, J., Sessa, V., Cooper, D., Julin, J., & Peyronnin, K. (1991). Some Differences Make a Difference: Individual Dissimilarity and Group Heterogeneity as Correlates of Recruitment, Promotions, and Turnover. *Journal of Applied Psychology, 76*, 675-689.

Jacobides, M. G. (2006). The Architecture and Design of Organizational Capabilities. *Industrial and Corporate Change, 15*, 151-171.

Janssen, M. (1993). *Microfoundations: A Critical Inquiry*. New York: Routledge.

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Administrative Science Quarterly, 24*, 602-611.

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). *Univariate Discrete Distributions*. Hoboken, NJ: John Wiley & Sons.

Jones, G. R. (1983). Psychological Orientation and the Process of Organizational Socialization: An Interactionist Perspective. *Academy of Management Review, 8*, 464-474.

Kane, A. A., Argote, L., & Levine, J. M. (2005). Knowledge Transfer between Groups via Personnel Rotation: Effects of Social Identity and Knowledge Quality. *Organizational Behavior and Human Decision Processes, 96*, 56-71.

Kane, G. C., & Alavi, M. (2007). Information Technology and Organizational Learning: An Investigation of Exploration and Exploitation Processes. *Organization Science, 18*, 796-812.

Karim, S. (2006). Modularity in Organizational Structure: The Reconfiguration of Internally Developed and Acquired Business Units. *Strategic Management Journal, 27*, 799-823.

Karim, S., & Mitchell, W. (2000). Path-Dependent and Path-Breaking Change: Reconfiguring Business Resources Following Acquisitions in the U.S. Medical Sector, 1978-1995. *Strategic Management Journal, 21*, 1061-1081.

Katz, D., & Kahn, R. (1978). *The Social Psychology of Organizations*. New York: Wiley.

Katz, M., & Shapiro, C. (1985). Network Externalities, Competition, and Compatibility. *The American Economic Review, 75*, 424-440.

Kauffman, S. A. (1993). *The Origin of Order: Self-Organization and Selection in Evolution*. New York: Oxford University Press.

Kay, N. (2013). Rerun the Tape of History and QWERTY Always Wins. *Research Policy, 42*, 1175-1185.

Kemp, R., Rip, A., & Schot, J. (2001). Constructing Transition Paths through the Management of Niches. In R. Graud & P. Karnoe (Eds.), *Path Dependence and Creation* (pp. 127-147). Mahwah, NJ: Lawrence Erlbaum Associates.

Kenneth, K. F., & Fahrbach, K. (1999). Organization Culture as a Complex System: Balance and Information in Models of Influence and Selection. *Organization Science, 10*, 253-277.

Kiesler, S., & Sproull, L. (1982). Managerial Response to Changing Environments: Perspectives from Social Cognition. *Administrative Science Quarterly, 27*, 548-570.

Kim, T., & Rhee, M. (2009). Exploration and Exploitation: Internal Variety and Environmental Dynamism. *Strategic Organization, 7*, 11-41.

Kimberly, J. R. (1975). Environmental Constraints and Organizational Structure: A Comparative Analysis of Rehabilitation Organizations. *Administrative Science Quarterly, 20*, 1-9.

Kirsch, D. A. (1996). *The Electric Car and the Burden of History: Studies in Automotive Systems Rivalry in America, 1890-1996.* Stanford University.

Knoke, D., & Yang, S. (2008). *Social Network Analysis*. Thousand Oaks: Sage Publications Inc.

Koch, J. (2008). Strategic Paths and Media Management: A Path Dependency Analysis of the German Newspaper Branch of High Quality Journalism. *Schmalenbach Business Review, 60*, 51-74.

Koch, J., Eisend, M., & Petermann, A. (2009). Path Dependence in Decision-Making Processes: Exploring the Impact of Complexity under Increasing Returns. *BuR - Business Research, 2*, 67-84.

Kraatz, M. S. (1998). Learning by Association? Interorganizational Networks and Adaptation to Environmental Change. *Academy of Management Journal, 41*, 621-643.

Krugman, P. R. (1979). Increasing Returns, Monopolistic Competition, and International Trade. *Journal of International Economics, 9*, 469-479.

Krugman, P. R. (1998). An Increasing Returns Chronology. In http://www.pkarchive.org/theory/ArthurChron.html.

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.

Langer, A. (2011). *Consumer Path Dependence in High-Tech Markets*. Berlin: Freie Universität Berlin.

Lant, T. K., & Mezias, S. (1992). An Organizational Learning Model of Convergence and Reorientation. *Organization Science, 3*, 47-71.

Lant, T. K., Miliken, F. J., & Batra, B. (1992). The Role of Managerial Learning and Interpretation in Strategic Persistence and Reorientation: An Empirical Exploration. *Strategic Management Journal, 13*, 585-608.

Lave, C. A., & March, J. G. (1975). *An Introduction to Models in the Social Sciences*. New York: Harper & Row.

Lavie, D. (2006). Capability Reconfiguration: An Analysis of Incumbent Responses to Technological Change. *Academy of Management Review, 31*, 153-174.

Lazer, D., & Friedman, A. (2007). The Network Structure of Exploration and Exploitation. *Administrative Science Quarterly, 52*, 667-694.

Leibenstein, H. (1971). Bandwagon, Snob and Veblen Effects in the Theory of Consumers' Demand. In W. Breit & H. M. Hochmann (Eds.), *Readings in Microeconomics* (pp. 115-116). New York: Holt, Rinehart and Winston, Inc.

Leonard-Barton, D. (1992). Core Capabilities and Core Rigidities: A Paradox in Managing New Product Development. *Strategic Management Journal, 13*, 111-125.

Levinthal, D. A. (1991). Organizational Adaptation and Environmental Selection-Interrelated Processes of Change. *Organization Science, 2*, 140-145.

Levinthal, D. A. (1997). Adaptation on Rugged Landscapes. *Management Science*, 934-950.

Levinthal, D. A., & March, J. G. (1993). The Myopia of Learning. *Strategic Management Journal, 14*, 95-112.

Levitt, B., & March, J. G. (1988). Organizational Learning. *Annual Review of Sociology, 14*, 319-338.

Levitt, S. D., & List, J. A. (2007). What do Laboratory Experiments Measuring Social Preferences Reveal about the Real World? *The Journal of Economic Perspectives, 21*, 153-174.

Leydesdorff, L. (2002). The Complex Dynamics of Technological Innovation: A Comparison of Models using Cellular Automata. *System Research and Behavioral Science, 19*, 563-575.

Liebowitz, S. J., & Margolis, S. E. (1990). The Fable of the Keys. *Journal of Law and Economics, 22*, 1-26.

Liebowitz, S. J., & Margolis, S. E. (1994). Network Externality: An Uncommon Tragedy. *The Journal of Economic Perspectives, 8*, 133-150.

Liebowitz, S. J., & Margolis, S. E. (1995a). Are Network Externalities a New Market Failure. *Research in Law and Economics, 17*, 1-22.

Liebowitz, S. J., & Margolis, S. E. (1995b). Path Dependence, Lock-In, and History. *Journal of Law, Economics, & Organization, 11*, 205-226.

Liebowitz, S. J., & Margolis, S. E. (1995c). Policy and Path Dependence: From QWERTY to Windows 95. *Regulation, 18*, 981-998.

Liebowitz, S. J., & Margolis, S. E. (1996). Should Technology Choice be a Concern for Antitrust? *Harvard Journal of Law and Technology, 9*, 283-318.

Liebowitz, S. J., & Margolis, S. E. (1999). *Winners, Losers & Microsoft; Competition and Antitrust in High Technology*. Oakland, CA: Independent Institute.

Liebowitz, S. J., & Margolis, S. E. (2013). The Troubled Path of the Lock-In Movement. *Journal of Competition Law and Economics, 9*, 125-152.

Loch, C. H., & Huberman, B. A. (1999). A Punctuated-Equilibrium Model of Technology Diffusion. *Management Science, 45*, 160-177.

Lomi, A., & Larsen, E. (1996). Interacting Locally and Evolving Globally: A Computational Approach to the Dynamics of Organizational Populations. *Academy of Management Journal, 39*, 1287-1321.

Lorscheid, I., Heine, B. O., & Meyer, M. (2011). Opening the 'Black Box' of Simulations: Increased Transparency of Simulation Models and Effective Results Reports through the Systematic Design of Experiments. In *Management@TUHH* (pp. 1-39). Hamburg: Hamburg University of Technology.

Lucas, H. C., & Goh, J. M. (2009). Disruptive Technology: How Kodak Missed the Digital Photography Revolution. *The Journal of Strategic Information Systems, 18*, 46-55.

Macal, C. M., & North, M. J. (2010). Tutorial on Agent-Based Modelling and Simulation. *Journal of Simulation, 4*.

Macy, M. W., & Willer, R. (2002). From Factors to Actors: Computational Sociology and Agent-Based Modeling. *Annual Review of Sociology, 28*, 143-166.

Mahmoud, H. (2009). *Polya Urn Models*. Boca Raton, FL: Chapman & Hall/CRC.

Mahoney, J. (2000). Path Dependence in Historical Sociology. *Theory and Society, 29*, 507-548.

Mahoney, J. (2001). Path-Dependent Explanations of Regime Change: Central America in Comparative Perspective. *Studies in Comparative International Development, 36*, 111-141.

Malerba, F., Nelson, R., Orsenigo, L., & Winter, S. (2008). Public Policies and Changing Boundaries of Firms in a "History-Friendly" Model of the Co-Evolution of the Computer and Semiconductor Industries. *Journal of Economic Behavior & Organization, 67*, 355-380.

Mallach, R. (2013). *Pfadabhängigkeit in Geschäftsbeziehungen*. Berlin: Freie Universität Berlin.

Malmberg, A., & Maskell, P. (2006). Localized Learning Revisited. *Growth and Change, 37*, 1-18.

Mandal, A. (2001). Overcoming Path Dependence. In *19th International Conference of the System Dynamics Society* (Vol. 19, pp. 102). Atlanta.

Maney, K. (2003). *The Maverick and His Machine: Thomas Watson, Sr. and the Making of IBM*. New Jersey: John Wiley & Sons, Inc.

March, J. G. (1981). Footnotes to Organizational Change. *Administrative Science Quarterly, 26*, 563-577.

March, J. G. (1991). Exploration and Exploitation in Organizational Learning. *Organization Science, 2*, 71-87.

March, J. G. (2006). Rationality, Foolishness and Adaptive Intelligence. *Strategic Management Journal, 27*, 201-214.

Marshall, A. (1890). *Principles of Economics*. London: Macmillan.

Martin, R., & Sunley, P. (2010). The Place of Path Dependence in an Evolutionary Perspective on the Economic Landscape. *Handbook of Evolutionary Economic Geography*, 62-92.

McGrath, R. G., Ferrier, W. J., & Mendelow, A. L. (2004). Real Options as Engines of Choice and Heterogeneity. *Academy of Management Review, 29*, 86-101.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology, 27*, 415-444.

Meyer, T. (2012). *Path Dependence in Two-Sided Markets - A Simulation Study on Technological Path Dependence with an Application to Platform Competition in the Smartphone Industry*. Berlin: Freie Universität Berlin.

Miller, D., & Friesen, P. (1984). *Organization: A Quantum View*. Englewood Cliffs, NJ: Prentince-Hall.

Miller, K. D., & Lin, S. (2010). Different Truths in Different Worlds. *Organization Science, 21*, 97-114.

Miller, K. D., & Martignoni, D. (2011). Organizational Learning with Bounded Rationality. In *Organization Science Winter Conference*. Steamboat Springs, Colorado.

Miller, K. D., Zhao, M., & Calantone, R. J. (2006). Adding Interpersonal Learning and Tacit Knowledge to March's Exploration-Exploitation Model. *Academy of Management Journal, 49*, 709-726.

Mintzberg, H. (1979). *The Structuring of Organizations*. London: Prentice Hall.

Mooney, C. Z. (1997). *Monte Carlo Simulation*. London: Sage Publications Ltd.

Moorman, C., & Miner, A. S. (1997). The Impact or Organizational Memory on New Product Performance and Creativity. *Journal of Marketing Research, 34*, 91-106.

Morand, E., Toulemon, L., Pennec, S., Baggio, R., & Nillari, F. (2010). Demographic Modelling: The State of the Art. In *SustainCity Working Paper*. Paris: Ined.

Munir, K. A., & Matthew, J. (2004). Discontinuity and After: The Social Dynamics of Technology Evolution Dominance. *Organization Studies, 25*, 561-581.

Nelson, R. R., & Winter, S. G. (1982). *An Evolutionary Theory of Economic Change*. Cambridge, MA: Belknap Press.

Nickerson, J. A., & Zenger, T. R. (2002). Being Efficiently Fickle: A Dynamic Theory of Organizational Choice. *Organization Science, 13*, 547-566.

Nikolai, C., & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation, 12*, 1-2.

Niosi, J. (1999). Fourth-Generation R&D: From Linear Models to Flexible Innovation. *Journal of Business Research, 45*, 111-117.

Nohria, N., & Gulati, R. (1996). Is Slack Good or Bad for Innovation? *Academy of Management Journal, 39*, 1245-1264.

Nonaka, I., & von Krogh, G. (2009). Perspective—Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organizational Knowledge Creation Theory. *Organization Science, 20*, 635-652.

Nooteboom, B. (1992). Towards a Dynamic Theory of Transactions. *Journal of Evolutionary Economics, 2*, 281-299.

Nooteboom, B. (2007). Cognitive Distance in and between Communities and Firms: Where do Exploitation and Exploration take Place and how are they Connected? In *Discussion Paper*. Tilburg: Tilburg University.

Nystrom, P. C., & Starbuck, W. H. (1984). To Avoid Organizational Crises - Unlearn. *Organizational Dynamics, 12*, 53-65.

O'Reilly, C. (1983). The Use of Information in Organizational Decision Making: A Model and Some Propositions. *Research in Organizational Behavior, 5*, 103-139.

Orlikowski, W., & Yates, J. (1992). Genres of Organizational Communication: A Structurational Approach to Studying Communication and Media. *The Academy of Management Review, 17*, 299-326.

Page, S. E. (2007). *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*: Princeton University Press.

Page, S. E. (2012). Aggregation in Agent-Based Models of Economies. *Knowledge Engineering Review, 27*, 151-162.

Pandza, K., & Thorpe, R. (2009). Creative Search and Strategic Sense-Making: Missing Dimensions in the Concept of Dynamic Capabilities. *British Journal of Management, 20*, 118-131.

Parker, G. G., & van Alstyne, M. W. (2005). Two-Sided Network Effects: A Theory of Information Product Design. *Management Science, 51*, 1494-1504.

Pentland, B. T., Feldman, M. S., Becker, M. C., & Liu, P. (2012). Dynamics of Organizational Routines: A Generative Model. *Journal of Management Studies, 49*, 1484-1508.

Petermann, A., Klaussner, S., & Senf, N. (2012). Organizational Path Dependence: The Prevalence of Positive Feedback Economics in Hierarchical Organizations. In K. G. Troitzsch, M. Moehring & U. Lotzmann (Eds.), *European Conference on Modelling and Simulation* (pp. 721-730). Koblenz.

Pettigrew, A. M. (1987). Context and Action in the Transformation of the Firm. *Journal of Management Studies, 24*, 649-670.

Pfeffer, J. (1983). Organizational Demography. *Research in Organizational Behavior, 5*, 299-357.

Pfeffer, J., & Salancik, J. (1978). *The External Control of Organizations*. New York: Harper and Row.

Pierson, P. (2000). Increasing Returns, Path Dependence, and the Study of Politics. *The American Political Science Review, 94*, 251-267.

Polhill, J. G., Parker, D., Brown, D., & Grimm, V. (2008). Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change. *Journal of Artificial Societies and Social Simulation, 11*, 1-3.

Postrel, S. R. (1990). Competing Networks and Proprietary Standards: The Case of Quadraphonic Sound. *The Journal of Industrial Economics, 39*, 169-185.

Powell, W., & DiMaggio, P. (1991). *The New Institutionalism in Organization Analysis*. Chicago: University of Chicago Press.

Prahalad, C. K., & Hamel, G. (1990). The Core Competence of the Corporation. *Harvard Business Review, 68*, 78-91.

Puffert, D. J. (2002). Path Dependence in Spatial Networks: The Standardization of Railway Track Gauge. *Explorations in Economic History, 39*, 282-314.

Radner, R. (1986). Preface Based on "Radner on Marschak". In C. McCuire & R. Radner (Eds.), *Decision and Organization: A Volume in Honor of Jacob Marschak*. Minneapolis: University of Minnesota Press.

Reinstaller, A., & Hoelzl, W. (2009). Big Causes and Small Events: QWERTY and the Mechanization of Office Work. *Industrial and Corporate Change, 18*, 999-1031.

Richiardi, M., Leombruni, R., & Saam, N. J. (2006). A Common Protocol for Agent-Based Social Simulation. *Journal of Artificial Societies and Social Simulation, 9*, 1-15.

Rivkin, J. W. (2000). Imitation of Complex Strategies. *Management Science, 46*, 824-844.

Rivkin, J. W. (2001). Reproducing Knowledge: Replication without Imitation at Moderate Complexity. *Organization Science, 12*, 274-293.

Rodan, S. (2005). Exploration and Exploitation Revisited: Extending March's Model of Mutual Learning. *Scandinavian Journal of Management, 21*, 407-428.

Rogers, E. M. (1962). *Diffusion of Innovation*. New York, NY: Free Press.

Rogers, E. M. (1991). The "Critical Mass" in the Diffusion of Interactive Technologies in Organizations. In K. L. Kraemer (Ed.), *The Information Systems Research Challenge: Survey Research Methods* (pp. 245-271). Boston, MA: Harvard Business School Press.

Romanelli, E., & Tushman, M. L. (1994). Organizational Transformation as Punctuated Equilibrium: An Empirical Test. *Academy of Management Journal, 37*, 1141-1166.

Rosenkopf, L., & Nerkar, A. (2001). Beyond Local Search: Boundary-Spanning, Exploration, and Impact in the Optical Disk Industry. *Strategic Management Journal, 22*, 287-306.

Ross, J., & Straw, B. M. (1993). Organizational Escalation and Exit: Lessons from the Shorehand Nuclear Power Plant. *Academy of Management Journal, 36*, 701-732.

Ross, S. M. (2009). *Introduction to Probability Models*. Burlington, MA: Academic press.

Sargent, R. G. (2010). Verification and Validation of Simulation Models. In B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan & E. Yuecesan (Eds.), *Winter Simulation Conference* (pp. 166-183). Baltimore, MD.

Sarkis, J., Gonzalez-Torre, P., & Adenso-Diaz, B. (2010). Stakeholder Pressure and the Adoption of Environmental Practices: The Mediating Effect of Training. *Journal of Operations Management, 28*, 163-176.

Sawyer, R. K. (2003). Artificial Societies: Multiagent Systems and the Micro-Macro Link in Sociological Theory. *Sociological Methods & Research, 31*, 325-363.

Schein, E. H. (2010). Three Cultures of Management: The Key to Organizational Learning. *"Glocal" working. Living and Working across the World with Cultural Intelligence*, 37-59.

Schelling, T. (1971). Dynamic Models of Segregation. *Journal of Mathematical Sociology, 1*, 143-186.

Schilling, J., & Kluge, A. (2009). Barriers to Organizational Learning: An Integration of Theory and Research. *International Journal of Management Reviews, 11*, 337-360.

Schilling, M. A. (2002). Technology Success and Failure in Winner-Take-All Markets: The Impact of Learning Orientation, Timing, and Network Externalities. *Academy of Management Journal, 45*, 387-398.

Schneider, A. L. (2006). Patterns of Change in the Use of Imprisonment in the American States: An Integration of Path Dependence, Punctuated Equilibrium and Policy Design Approaches. *Political Research Quarterly, 59*, 457-470.

Schneider, B., Goldstein, H., & Smith, B. (1995). The ASA Framework: An Update. *Personnel Psychology, 48*, 747-773.

Schneider, M., & Somers, M. (2006). Organizations as Complex Adaptive Systems: Implications of Complexity Theory for Leadership Research. *The Leadership Quarterly, 17*, 351-365.

Schulte, B. (2013). Staying the Consumption Course - Exploring Lock-In Processes under Interdependence. In G. Oderkeken, J. Field & A. Smart (Eds.), *Service Management and Science Forum 2013*. Las Vegas.

Schulze, A., & Brojerdi, G. (2012). The Effect of the Distance between Partners' Knowledge Components on Collaborative Innovation. *European Management Review, 9*, 85-98.

Schumacker, R. E., & Lomax, R. G. (2010). *A Beginner's Guide to Structural Equation Modeling*. London: Routledge.

Schwaninger, M., & Mandl, C. (2012). Understanding the System Dynamics of High-Technology Markets: Pólya Processes with Positive Feedback, Path Dependence and Lock-In. In *Eurocast 2011* (Vol. 6927, pp. 129-136). Las Palmas de Gran Canaria.

Schwartz, H. (2004). *Down the Wrong Path: Path Dependence, Increasing Returns, and Historical Institutionalism.* Unpublished Manuscript, University of Virginia, Charottesville, VA.

Seidel, E. (2012). *Path Dependence and the Environmental Context - An Inquiry into the Effects of Environmental Complexity and Turbulence on Path Dependence in Organizational Learning*. Berlin: Freie Universität Berlin.

Setterfield, M. (1999). Expectations, Path Dependence and Effective Demand: A Macroeconomic Model along Keynesian Lines. *Journal of Post Keynesian Economics, 21*, 479-501.

Siggelkow, N. (2001). Change in the Presence of Fit: The Rise, the Fall, and the Renaissance of Liz Claiborne. *Academy of Management Journal, 44*, 838-857.

Siggelkow, N., & Rivkin, J. W. (2009). Hiding the Evidence of Valid Theories: How Coupled Search Processes Obscure Performance Differences among Organizations. *Administrative Science Quarterly, 54*, 602-634.

Silvestre, J. (1987). *Economies and Diseconomies of Scale* (Vol. 2). London: Macmillan and Stockton.

Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society, 106*, 467-482.

Simon, H. A. (1964). On the Concept of Organizational Goal. *Administrative Science Quarterly, 9*, 1-22.

Simon, H. A. (1972). Theories of Bounded Rationality. *Decision and Organization, 1*, 161-176.

Smil, V. (2006). *Transforming the Twentieth Century: Technical Innovations and their Consequences*. New York: Oxford University Press.

Sood, A., & Tellis, G. J. (2011). Demystifying Disruption: A New Model for Understanding and Predicting Disruptive Technologies. *Marketing Science, 30*, 339-354.

Spell, C. S. (1999). Where do Management Fashions come from, and how long do they stay? *Journal of Management History (Archive), 5*, 334-348.

Starbuck, W. H. (1996). Unlearning Ineffective or Obsolete Technologies. In *NYU Working Paper Nr. 2451/14188*. New York: New York University.

Sterman, J., Fiddaman, T., Franck, T., Jones, A., McCauley, S., Rice, P., Sawin, E., & Siegel, L. (2013). Management Flight Simulators to Support Climate Negotiations. *Environmental Modelling & Software, 44*, 122-135.

Sterman, J. D. (2000). *Business Dynamics - Systems Thinking and Modelling for a Complex World*. New York: McGraw-Hill.

Sterman, J. D., & Wittenberg, J. (1999). Path Dependence, Competition, and Succession in the Dynamics of Scientific Revolution. *Organization Science, 10*, 322-341.

Stieglitz, N., & Heine, K. (2007). Innovations and the Role of Complementarities in a Strategic Theory of the Firm. *Strategic Management Journal, 28*, 1-15.

Stinchcombe, A. L. (1965). *Social Structure and Organizations*. New York: Rand McNally.

Suarez, F. F., & Lanzolla, G. (2007). The Role of Environmental Dynamics in Building a First Mover Advantage Theory. *Academy of Management Review, 32*, 377-392.

Suarez, F. F., & Oliva, R. (2005). Environmental Change and Organizational Transformation. *Industrial and Corporate Change, 14*, 1017-1041.

Sundararajan, A. (2008). Local Network Effects and Complex Network Structure. *The B.E. Journal of Theoretical Economics, 7*, 1935-1704.

Sydow, J., Schreyögg, G., & Koch, J. (2005). Organizational Paths: Path Dependency and Beyond. In *21st EGOS Colloquium*. Berlin.

Sydow, J., Schreyögg, G., & Koch, J. (2009). Organizational Path Dependence: Opening the Black Box. *Academy of Management Review, 34*, 689-709.

Terborg, J. R., & Lee, T. W. (1984). A Predictive Study of Organizational Turnover Rates. *The Academy of Management Journal, 27*, 793-810.

Thompson, J. (1967). *Organizations in Action*. New York: McGraw-Hill.

Tidd, J., & Bessant, J. (2009). *Managing Innovation: Integrating Technological, Market and Organizational Change* (4 ed.). London: John Wiley & Sons Ltd.

Tripsas, M., & Gavetti, G. (2000). Capabilities, Cognition and Inertia: Evidence from Digital Imaging. *Strategic Management Journal, 21*, 1147-1161.

Tushman, M. L., & Anderson, P. (1986). Technological Discontinuities and Organizational Environments. *Administrative Science Quarterly, 31*, 439-465.

Tushman, M. L., Newman, W. H., & Romanelli, E. (1986). Convergence and Upheaval: Managing the Unsteady Pace of Organizational Evolution. *California Management Review, 29*, 583-594.

Tushman, M. L., & Romanelli, E. (1985). Organizational Evolution: A Metamorphosis Model of Convergence and Reorientation. In L. L. Cummings & E. M. Staw (Eds.), *Research in Organizational Behavior* (Vol. 7, pp. 171-222). Greenwich, CT: JAI Press.

Uotila, J., Maula, M., Keil, T., & Zahra, S. A. (2009). Exploration, Exploitation, and Financial Performance: Analysis of S&P 500 Corporations. *Strategic Management Journal, 30*, 221-231.

Van Den Bosch, F. A. J., Volberda, H. W., & de Boer, M. (1999). Coevolution of Firm Absorptive Capacity and Knowledge Environment: Organizational Forms and Combinative Capabilities. *Organization Science, 10*, 551-568.

Van Horn, R. L. (1971). Validation of Simulation Results. *Management Science, 17*, 247-258.

Van Maanen, J. (1979). The Fact of Fiction in Organizational Ethnography. *Administrative Science Quarterly, 24*, 539-550.

Vergne, J. P. (2013). QWERTY is Dead, Long Live Path Dependence! *Research Policy, 42*, 1191-1194.

Vergne, J. P., & Durand, R. (2010). The Missing Link between the Theory and Empirics of Path Dependence: Conceptual Clarification, Testability Issue and Methodological Implications. *Journal of Management Studies, 47*, 736-759.

Volberda, H. W. (1996). Toward the Flexible Form: How to Remain Vital in Hypercompetitive Environments. *Organization Science, 7*, 359-374.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. Chicago/IL: University of Illinois Press.

Voss, G. B., Sirdeshmukh, D., & Voss, Z. G. (2008). The Effects of Slack Resources and Environmental Threat on Product Exploration and Exploitation. *Academy of Management Journal, 51*, 147-164.

Waddock, S., & Graves, S. (1997). The Corporate Social Performance-Financial Performance Link. *Strategic Management Journal, 18*, 303-319.

Walsh, J. P., & Ungson, G. R. (1991). Organizational Memory. *Academy of Management Review, 16*, 57-91.

Webster, M., & Sell, J. (2007). *Laboratory Experiments in the Social Sciences*. Burlington, MA: Elsevier Inc.

Weick, K. E., & Roberts, K. H. (1993). Collective Mind in Organizations: Heedful Interrelating on Flight Decks. *Administrative Science Quarterly, 38*, 357-381.

Wiersema, M. F., & Bantel, K. A. (1992). Top Management Team Demography and Corporate Strategic Change. *Academy of Management Journal, 35*, 91-121.

Wiersema, M. F., & Bantel, K. A. (1993). Top Management Team Turnover as an Adaptation Mechanism: The Role of the Environment. *Strategic Management Journal, 14*, 485-504.

Williams, K. Y., & O'Reilly, C. A. (1998). Demography and Diversity in Organizations: A Review of 40 Years of Research. *Research in Organizational Behavior, 20*, 77-140.

Williams, R. (2004). Management Fashions and Fads: Understanding the Role of Consultants and Managers in the Evolution of Ideas. *Management Decision, 42*, 769-780.

Witt, U. (1997). "Lock-In" vs. "Critical Masses" - Industrial Change under Network Externalities. *International Journal of Industrial Organization, 15*, 753-773.

Wolcott, R. C., & Lippitz, M. J. (2007). The Four Models of Corporate Entrepreneurship. *MIT Sloan Management Review, 49*, 74-82.

Wong, S. (2004). Distal and Local Group Learning: Performance Trade-Offs and Tensions. *Organization Science, 15*, 645-656.

Woolridge, M., & Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. *Knowledge Engineering Review, 10*, 115-152.

Yin, R. K. (2009). *Case Study Research: Design and Methods* (4 ed.). Thousand Oaks: Sage Inc.

Zajac, E. J., & Kraatz, M. S. (1993). A Diametric Forces Model of Strategic Change: Assessing the Antecedents and Consequences of Restructuring in the Higher Education Industry. *Strategic Management Journal, 14*, 83-102.

Zajac, E. J., & Kraatz, M. S. (2007). A Diametric Forces Model of Strategic Change: Assessing the Antecedents and Consequences of Restructuring in the Higher Education Industry. *Strategic Management Journal, 14*, 83-102.

Zelditch, M. (1969). *Can you really Study an Army in the Laboratory*. New York: Holt, Rinehart and Winston.

A. Appendices

In the following the code for the simulations is provided. The original files can be downloaded from http://pastebin.com/u/FelixOb. For the sake of clarity all the code needed to replicate the simulation experiments is included.

A.1 Replication of the March Model

```java
/*
 * MarchModel.java
 *
 * (c) 2013, Felix Obschonka
 */
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.Random;
import java.util.Vector;

public class MarchModel {
	private Vector<Integer> code;
	private Vector<Integer> reality;
	private Vector<Vector<Integer>> levelOrgOriginal;
	private Vector<Vector<Integer>> levelOrgGuess;
	public static void main(String[] args) {
		MarchModel sim = new MarchModel();
		/*
		 * Defines Output-Type for FileWriter:
		 *    1: Average individual knowledge of organization
		 *    2: Knowledge of organizational code
		 */
		int outputtype = 2;
		// Parameters for the simulation
		double P1 = 0.1; // Learning agents from code
		double P2 = 0.9; // Learning code from dominant agents
		double P3 = 0.0; // Turnover rate
		double P5 = 0.0; // Turbulence rate

		// Starts simulation
		double result = sim.run(P1, P2, P3, P5, outputtype);
		result = Math.round(result * 1000.0) / 1000.0;
		// Writes the results in file
		String outname = "output.txt";
		switch (outputtype) {
```

```
        case 1:
              outname = "avgIndOrg.txt";
              break;
        case 2:
              outname = "codeKnowl.txt";
              break;
        default:
        }
        try {
              FileWriter outfile = new FileWriter(outname, true);
              PrintWriter out = new PrintWriter(outfile);
              out.println("" + result);
              out.close();
              outfile.close();
        } catch (Exception x) {
              System.out.println("FileWriter error");
        }
  }
  public double run(double P1, double P2, double P3,
                          double P5, int outputtype) {
  int timeslots = 100;        // Simulation run time in ticks
  int numberOfAgents = 50;    // Number of agents
  int numIt = 50;             // Number of iterations
  int sizeAgents = 30;        // Size of agents
  // Data structures containing knowledge of entities
  Vector<Double> averageIndKnowledge = new
  Vector<Double>(timeslots);
  Vector<Double> codeKnowledge = new Vector<Double>(timeslots);
  Vector<Double> sumIndKnowledge = new Vector<Double>(timeslots);
  Vector<Double> sumCodeKnowledge = new Vector<Double>(timeslots);
        for (int i = 0; i < timeslots; i++) {
              sumIndKnowledge.add(0.0);
              sumCodeKnowledge.add(0.0);
        }
        // Loop over iterations
        for (int it = 0; it < numIt; it++) {
              // Initialize data structures
              code = new Vector<Integer>(sizeAgents);
              for (int i = 0; i < sizeAgents; i++) {
                    code.add(0);
              }
              reality = generateReality(sizeAgents);
              levelOrgOriginal =
              generate(sizeAgents, numberOfAgents);
              Random rnd = new Random();
              for (int t = 0; t < timeslots; t++) {
              // Create duplicate of organization
              levelOrgGuess =
              new Vector<Vector<Integer>>(numberOfAgents);
              for (int m = 0; m < numberOfAgents; m++) {
                    Vector<Integer> originalAgent =
```

```
                levelOrgOriginal.get(m);
                Vector<Integer> guessAgent =
                new Vector<Integer>(originalAgent);

            for (int n = 0; n < sizeAgents; n++) {
                int agentValue =
            originalAgent.get(n);
                    if (agentValue == 0) {
                        if (rnd.nextDouble() < 0.5) {
                            agentValue = -1;
                        } else {
                            agentValue = 1;
                            }
                        guessAgent.set(n, agentValue);
                        }
                    }
                    levelOrgGuess.add(guessAgent);
                }
        // Performance measurements
        double currentIndKnowledgeOrg =
        averageIndKnowledge(levelOrgGuess, reality);
        sumIndKnowledge.set(t, sumIndKnowledge.get(t)
        + currentIndKnowledgeOrg);
        double currentCodeKnowledgeOrg = codeKnowledge(code, reality
        sumCodeKnowledge.set(t, sumCodeKnowledge.get(t)
        + currentCodeKnowledgeOrg);
        // Procedure of simulation
        learnOrgfromCode(P1);      // 1. Agents learn from code
        learnCodefromOrg(P2);      // 2. Code learns from agents
        turbulenceReality(P5);     // 3. Environmental change
        turnover(P3, sizeAgents); // 4. Turnover of agents
            }
        }
        // Average knowledge for each timeslot over iterations
        for (int t = 0; t < timeslots; t++) {
            averageIndKnowledge.add
            (sumIndKnowledge.get(t) / numIt);
            codeKnowledge.add(sumCodeKnowledge.get(t) / numIt);
            System.out.println("t=" + t + " Average Individual
            Knowledge: "+ averageIndKnowledge.get(t) + " Code
            Knowledge: "+ codeKnowledge.get(t));
        }
        if (outputtype == 2)
            return averageIndKnowledge.get(timeslots - 1);
        else
            return codeKnowledge.get(timeslots - 1);
    }
    // Method for agents learning from code
    private void learnOrgfromCode(double P1) {
        for (int i = 0; i < levelOrgOriginal.size(); i++) {
            Vector<Integer> currentAgent =
```

```
                levelOrgOriginal.get(i);
                learnnotzero(currentAgent, code, P1);
        }
}
// Method for code learning from agents
private void learnCodefromOrg(double P2) {
        Vector<Integer> majority = new Vector<Integer>(code.size());
        for (int j = 0; j < code.size(); j++) {
                majority.add(0);
        }
        for (int i = 0; i < levelOrgGuess.size(); i++) {
                Vector<Integer> currentAgent = levelOrgGuess.get(i);
                if (isDominant(currentAgent, code, reality)) {
                        for (int j = 0; j < code.size(); j++) {
                                majority.set(j, majority.get(j) +
                                currentAgent.get(j));
                        }
                }
        }
Random rnd = new Random();
for (int i = 0; i < code.size(); i++) {
        if (code.get(i) * majority.get(i) <= 0 &&
        majority.get(i) != 0) {
                int k = Math.abs(majority.get(i));
                double probChange = 1 - Math.pow((1 - P2), k);
                double currentProb = rnd.nextDouble();
                        if (currentProb <= probChange) {
                                if (majority.get(i) > 0) {
                                        code.set(i, 1);
                                } else {
                                        code.set(i, -1);
                                }
                        }
                }
        }
}
// Method for turnover of agents
private void turnover(double P5, int sizeAgents) {
        Random rnd = new Random();
        for (int i = 0; i < levelOrgOriginal.size(); i++) {
                double currentProb = rnd.nextDouble();
                if (currentProb <= P5) {
                Vector<Integer> newAgent = generateAgent(sizeAgents);
                levelOrgOriginal.set(i, newAgent);
                }
        }
}
// Method for turbulence in reality
private void turbulenceReality(double PR1) {
        Random rnd = new Random();
        for (int i = 0; i < reality.size(); i++) {
```

```
                double currentProb = rnd.nextDouble();
                if (currentProb <= PR1) {
                    if (reality.get(i) == 1) {
                        reality.set(i, -1);
                    } else {
                        reality.set(i, 1);
                    }
                }
            }
        }
        // Method to check for dominant agents
        private boolean isDominant(Vector<Integer> testAgent,
        Vector<Integer> code, Vector<Integer> reality) {
            boolean result = false;
                if (compareAgentSum(testAgent, reality) >
                    compareAgentSum(code, reality)) {
                    result = true;
            }
            return result;
        }
        // Learning method
        private void learnnotzero(Vector<Integer> vector1,
            Vector<Integer> vector2, double p) {
            Random rnd = new Random();
            for (int i = 0; i < vector1.size(); i++) {
                if (vector2.get(i) == 0) {
                    continue;
                }
                double currentProb = rnd.nextDouble();
                if (currentProb < p) {
                    if (vector1.get(i) * vector2.get(i) == -1) {
                        vector1.set(i, 0);
                    } else if (vector1.get(i) == 0) {
                        vector1.set(i, vector2.get(i));
                    }
                }
            }
        }
        // Method for generating organization
        private Vector<Vector<Integer>> generate(int sizeAgents, int
                num)
        {
        Vector<Vector<Integer>> result = new Vector<Vector<Integer>>(num);
            for (int i = 0; i < num; i++) {
                Vector<Integer> agent = generateAgent(sizeAgents);
                result.add(agent);
        }
            return result;
        }
        // Method for generating random agent
        private Vector<Integer> generateAgent(int sizeAgents) {
```

```
        Vector<Integer> agent = new Vector<Integer>(sizeAgents);
        Random rnd = new Random();
        for (int j = 0; j < sizeAgents; j++) {
              double p = rnd.nextDouble();
              int value = (int) Math.floor(p * 3) - 1;
              agent.add(value);
        }
        return agent;
  }
  // Generate and initialize reality vector
  private Vector<Integer> generateReality(int sizeReality) {
        Vector<Integer> result = new Vector<Integer>(sizeReality);
        Random rnd = new Random();
        for (int i = 0; i < sizeReality; i++) {
              double p = rnd.nextDouble();
              int value = ((int) Math.round(p)) * 2 - 1;
              result.add(value);
        }
        return result;
  }
  // Method for comparing agent with organizational code
  private int compareAgentSum(Vector<Integer> agent,
        Vector<Integer> code) {
        int equalsAgentsSum = 0;
        for (int i = 0; i < code.size(); i++) {
              equalsAgentsSum += agent.get(i) * code.get(i);
        }
        return equalsAgentsSum;
  }
  // Measure average individual knowledge
  private double averageIndKnowledge(Vector<Vector<Integer>>
  agentLevel, Vector<Integer> reality) {
        int totalKnowledge = 0;
        for (int i = 0; i < agentLevel.size(); i++) {
              Vector<Integer> currentAgent = agentLevel.get(i);
              totalKnowledge = totalKnowledge
              + compareAgentSum(currentAgent, reality);
        }
        double average = (double) totalKnowledge
        / ((double) agentLevel.size() * (double) reality.size());
        return average;
  }
  // Measure code knowledge
  private double codeKnowledge(Vector<Integer> code,
        Vector<Integer> reality) {
  return (double) compareAgentSum(reality, code) / (double)
  reality.size();
  }
}//END
```

A.2 Interpersonal Organizational Learning Model

The intra-organizational learning model consists of the runnable simulation model in which the parameters for conducting the experiments are altered and methods are called. The model uses the classes described in Table 15.

Table 15: Java classes used for the simulation

Java Class	Remarks	Chapter
Grid	Contains methods to construct the organizational grid, turnover, rotation and environmental change	A.2.1
Agent	Describes the characteristics of agents	A.2.2
Environment	Describes the characteristics of the environment	A.2.3
SimilarityAgent	Contains the logic for learning in accordance with similarity	A.2.4
GridAgent	Contains the logic for selection and organizational learning	A.2.5

A.2.1 Simulation model ("OrgLearningModel.java")

```java
/*
 *  OrgLearningModell.java
 *
 *  (c) 2013, Felix Obschonka
 *
 */

package learningAgents.gridNetworking;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Random;
import java.util.Vector;
import learningAgents.Agent;

public class OrgLearningModell {

public static void main(String[] args) {

/* -----------------------------------------------------
 * - Settings for intra-organizational learning model -
 * -----------------------------------------------------
 */

int steps          = 300;     // Number of steps per run
int iterations     = 300;     // Number of iterations
int gridSize       = 10;      // Length & width of grid
int size           = 75;      // Number of dimensions for agent

/* -------------------------------
 * - Settings for the base model -
 * -------------------------------
 */

double similarityWeight = 0.0; // States similarity parameter
double paramY = 1 - similarityWeight;
```

```java
/* -----------------------------------------------------
 * - Settings for the first extension: Exogenous shock -
 * -----------------------------------------------------
 */

double turbulenceRate = 1.0;            // Chance of turbulence
int pointOfSignificantChange = 150;     // Time step of change

/*
 * ----------------------------------------------------------
 * - Settings for the second extension: Turnover & Rotation -
 * ----------------------------------------------------------
 */

double turnoverRate = 0.0; // Chance for turnover of agent
double rotateRate = 0.00;  // Chance for rotation of agent

/* Turnover:
 *     true:  Agents are randomly replaced
 *     false: Agents are replaced according to turnoverType
 */
boolean turnover = false;

/* Type of Turnover:
 *     true:  Only agents at group borders are turned over
 *     false: Only agents within groups are turned over
 */
boolean turnoverType = false;

/*
 * ---------------------------------------------------------
 * - Settings for the third extension: Top Management Team -
 * ---------------------------------------------------------
 */

// Percentage of agents in the organization assigned to the TMT
double bestPercentage = 0.05;
// Probability for learning from TMT before change
double learningTMTbefore = 0.01;
// Probability for learning from TMT after change
double learningTMTafter  = 0.01;
// Probability of replacing agent(s) in the TMT
double turnoverTMT = 1.0;
// Number of agents replaced in the TMT
int turnoverTMTnum = 1;
// At this step the TMT is formed
int pointOfTMTForming = 0;
// From this step onwards agents learn from TMT
int startLearningFromTMT = 151;

/* TMT Agents get replaced by:
```

```java
 *    true:  By the best agents within the organization
 *    false: By random agents from outside the organization
 */
boolean takeBestforTMT = false;

/*
 * Turnover timing of TMT
 *    true:  Turnover only once at point of environmental change
 *    false: Turnover every time step
 */
boolean turnoverTMTOnce = true;

/* ----------------
 * - Measurements -
 * ----------------
 */

/* Performance Output:
 *   true:  Output of last performance values for each run
 *   false: Output of the mean performance over iterations
 */
boolean showLastPerformanceValues = false;

// Stores changes in interactions after environmental change
ArrayList<Integer> learningInteractionChanged = new
ArrayList<Integer>();

// Defines the performance threshold for delocking
double threshhold = 0.1;

/* This is the point where measurements will take place that are
 * important for knowing how the grid looks BEFORE the shock
 */
int beforePointOfSignificantChange = 20;

/* This is the point where measurements will take place that are
 * important for knowing how the grid looks AFTER the shock
 */
int afterPointOfSignificantChange = 130;

/* Initialization of agents in the organization:
 *    true:  Every iteration has the same agents at set-up
 *    false: Random agents for every run
 */
boolean sameAgentsForEveryIteration = false;
// Number of runs exceeding the threshold for unlocking
int recoveredCounter = 0;

final Vector<Double> lastPerformanceValues = new
Vector<Double>(iterations);
final Vector<Double> sumAvgIndKnowledge = new Vector<Double>(steps);
```

```
      for (int i = 0; i < steps; i++) {
      sumAvgIndKnowledge.add(0.0);
      }

// Initialize data structure for groups
List<Integer> groupsBeforeShock =
new ArrayList<Integer>(iterations);
List<Integer> groupsAfterShock =
new ArrayList<Integer>(iterations);
List<Integer> groupsEndOfTicks =
new ArrayList<Integer>(iterations);
Map<Integer, Integer> actualSwaps =
new HashMap<Integer,   Integer>();

GridAgent tmtAgent = new
GridAgent(size, 0, 0, null, 0, false);

Random rnd = new Random();
boolean singleRun = iterations == 1;

// Initialize grid
grid = new Grid(gridSize, size, turbulenceRate, turnoverRate, rotateRate,
0.0);

/*
 * Make an exact copy of all initial agents. This is needed if
 * every iteration should start with the same agents.
 */
Vector<Vector<ArrayList<Integer>>> initialKnowledge = new
Vector<Vector<ArrayList<Integer>>>();
      if(sameAgentsForEveryIteration){
      Vector<Vector<GridAgent>> agents = grid.getAgents();
            for(int i = 0; i < agents.size(); i++){
           Vector<ArrayList<Integer>> knowRow = new
           Vector<ArrayList<Integer>>();
                  for(int j = 0; j < agents.get(i).size(); j++){
                  GridAgent a = agents.get(i).get(j);
                  ArrayList<Integer> know = new
                  ArrayList<Integer>();
                        for(int k : a.getKnowledge()){
                        know.add(k);
                        }
                        knowRow.add(know);
                  }
                  initialKnowledge.add(knowRow);
            }
      }

// Start of iterations
for (int it = 0; it < iterations; it++) {
      boolean recovered = false;
```

```
        ArrayList<Integer> tmtKnowledge = new
        ArrayList<Integer>();
        for (int i = 0; i < size; i++) {
              tmtKnowledge.add(0);
              }

              // Checks for same agent assumption
              if (sameAgentsForEveryIteration) {
                    Vector<Vector<GridAgent>> agents =
                    grid.getAgents();
                    for(int i = 0; i < agents.size(); i++){
                    for(int j = 0; j < agents.get(i).size();
                    j++){
                                ArrayList<Integer> know =
                                initialKnowledge.get(i).get(j);
                                GridAgent a = grid.getAgent(i, j);
                                ArrayList<Integer> newKnow = new
                                ArrayList<Integer>();
                                      for(int e : know){
                                            newKnow.add(e);
                                      }
                                      a.setKnowledge(newKnow);
                                }
                          }
                    } else {
                          grid = new Grid(gridSize, size, turbulenceRate,
                                      turnoverRate, rotateRate, 0.0);
                    }

// Initialize TMT
List<Agent> tmt = null;
for (int t = 0; t < steps; t++) {

        // Turnover TMT at point of significant change
        if(t==pointOfSignificantChange && turnoverTMTOnce) {
              if(turnoverTMTOnce){


              // Replace Agents of TMT with agents of the organization
              List<Agent> gAgents = new
              ArrayList<Agent>(grid.getAgentsList());
                    if (takeBestforTMT) {

                    // Sort agents descending by accordance to reality
                    Collections.sort(gAgents, new Comparator<Agent>() {
                    @Override
                    public int compare(Agent agent0, Agent agent1) {
                    int know0=agent0.getAccordanceValue
                    (grid.getReality());
                    int know1 =
                    agent1.getAccordanceValue(grid.getReality());
```

```
                if (know0 > know1) {
                        return 1;
                        }
                        if (know0 < know1) {
                                return -1;
                        }
                                return 0;
                        }
        });
        int index = 0;
        List<Agent> swapList = new ArrayList<Agent>();
        for(int j = 0; j < turnoverTMTnum; j++){
                while(tmt.contains(gAgents.get(index))){
                        index++;
                        }

                swapList.add(tmt.get(index));
                }

                // Only swap agents that haven't been swapped
                List<Integer> alreadySwapped = new
                ArrayList<Integer>();
                        for(int i = 0; i < turnoverTMTnum; i++){
                        int randomNumber = 0;
                                do{
                                randomNumber = rnd.nextInt(tmt.size());
                                }
                while(alreadySwapped.contains(randomNumber));

                alreadySwapped.add(randomNumber);
                tmt.set(randomNumber, swapList.remove(0));
                                }
                        } else {


                        // Replace TMT agents with new agents
                        Collections.shuffle(gAgents);
                        List<Integer> alreadySwapped =
                        new ArrayList<Integer>();
                                for(int i = 0; i < turnoverTMTnum; i++){
                                int randomNumber = 0;
                                        do{
                                randomNumber = rnd.nextInt(tmt.size());
                                                        }
                        while(alreadySwapped.contains(randomNumber));
                        Agent oldAgent = tmt.get(randomNumber);

oldAgent.setKnowledge(generateKnowledge(size));

alreadySwapped.add(randomNumber);
                                                }
```

```
                        }
                    }
                }
    if(t == pointOfTMTForming){

    // TMT is formed at this point of time
    tmt = formTMT(bestPercentage);
        }
        if (t >= pointOfTMTForming) {
            if(rnd.nextDouble() < turnoverTMT){

                // TMT turnover
                if(!turnoverTMTOnce){

                    // Replace agents of TMT
                    List<Agent> gAgents = new
                    ArrayList<Agent>(
                    grid.getAgentsList());
                    if (takeBestforTMT) {

                    // Sort agents descending by accordance
                    Collections.sort(gAgents, new
                    Comparator<Agent>() {

@Override
public int compare(Agent agent0, Agent agent1) {
    int know0 = agent0.getAccordanceValue(grid.getReality());
    int know1 = agent1.getAccordanceValue(grid.getReality());
        if (know0 > know1) {
            return 1;
            }
            if (know0 < know1) {
                return -1;
                    }
            return 0;
                }

            });
int index = 0;
List<Agent> swapList = new ArrayList<Agent>();
for(int j = 0; j < turnoverTMTnum; j++){
    while(tmt.contains(gAgents.get(index))){
        index++;
        }
    swapList.add(tmt.get(index));
    }

    // Only swap agents that haven't been swapped before
    List<Integer> alreadySwapped = new ArrayList<Integer>();
        for(int i = 0; i < turnoverTMTnum; i++){
            int randomNumber = 0;
```

```
                  do{
                  randomNumber = rnd.nextInt(tmt.size());
                              }
                        while(alreadySwapped.contains(randomNumber));
                        alreadySwapped.add(randomNumber);
                        tmt.set(randomNumber, swapList.remove(0));
                        }
                  } else {

                  // Replace TMT agents with completely new agents
                  Collections.shuffle(gAgents);
                  List<Integer> alreadySwapped =
                  new ArrayList<Integer>();
                  for(int i = 0; i < turnoverTMTnum; i++){
                        int randomNumber = 0;
                        do{
                        randomNumber = rnd.nextInt(tmt.size());
                                    }
                        while(alreadySwapped.contains(randomNumber));
                        Agent oldAgent = tmt.get(randomNumber);
                        oldAgent.setKnowledge(generateKnowledge(size));
                        alreadySwapped.add(randomNumber);
                                                }
                                          }
                                    }
                              }



// Calculate current opinion of TMT
for (int i = 0; i < tmtKnowledge.size(); i++) {
      tmtKnowledge.set(i, 0);
      }
      for (Agent tmtMember : tmt) {
          for (int i = 0; i < tmtMember.getKnowledge().size(); i++){
      tmtKnowledge.set(i, tmtKnowledge.get(i)+
      tmtMember.getKnowledge().get(i));
                                    }
                              }
            for (int i = 0; i < tmtKnowledge.size(); i++) {
                  if (tmtKnowledge.get(i) > 0) {
                        tmtKnowledge.set(i, 1);
                        } else if (tmtKnowledge.get(i) < 0) {
                        tmtKnowledge.set(i, -1);
                              } else {
                              tmtKnowledge.set(i, 0);
                                    }
                              }
                        tmtAgent.setKnowledge(tmtKnowledge);
                        }
```

```
if (t >= startLearningFromTMT) {

// Agents learn from TMT
      for (Agent agent : grid.getAgentsList()) {
            if(t<= pointOfSignificantChange)
            agent.learn(tmtAgent, learningTMTbefore);
            else
            agent.learn(tmtAgent, learningTMTafter);
                              }
                        }

// Agents learn from each other in a random order
for (GridAgent agent : rumbleList(grid.getAgentsList())) {
      SimilarityAgent currentAgent = (SimilarityAgent) agent;
      currentAgent.learnFromBestNeighbour(similarityWeight, paramY);
                        }

// Knowledge measurement
double currentKnowledge = grid.getCurrentAvgIndKnowledge();
sumAvgIndKnowledge.set(t, sumAvgIndKnowledge.get(t)
                                    + currentKnowledge);
if(singleRun){

if (t == pointOfSignificantChange-beforePointOfSignificantChange) {


      // Count groups
      List<Agent> sortedAgents =
      new ArrayList<Agent>(grid.getAgentsList());
      Collections.sort(sortedAgents);
      int groups = 1;
      int[][] groupGrid = new int[gridSize][gridSize];
            for (int i = 0; i < sortedAgents.size() - 1; i++) {
            GridAgent cAgent = (GridAgent) sortedAgents.get(i);

            groupGrid[cAgent.getX()][cAgent.getY()] = groups;
            if (sortedAgents.get(i).compareTo
            (sortedAgents.get(i+1)) != 0) {
                  groups++;
                                    }
                              }
                  groupsBeforeShock.add(groups);
                  GridAgent cAgent =
                  (GridAgent)sortedAgents.get(sortedAgents.size() -1);
            groupGrid[cAgent.getX()][cAgent.getY()] = groups;

      // Print visualization of grid and group membership
      for(int i = 0; i < groupGrid.length; i++){
            for (int j = 0; j < groupGrid[i].length; j++) {
                  System.out.print(groupGrid[i][j] + " ");
                        }
```

```
                System.out.println();
                }
                System.out.println();
                System.out.println();
                if(sameAgentsForEveryIteration){
                      for(int x = 0; x < grid.getX(); x++){
                                  String r1 = "";
                                  String r3 = "";
                                  String r2 = "";
                      for(int y = 0; y < grid.getY(); y++){
                      SimilarityAgent sAgent = (SimilarityAgent)
                      grid.getAgent(x, y);
                sAgent.calculateSimilarityToNeighbours();
                GridAgent downNeighbour =
                grid.getAgent((x >= grid.getX()-1)?0:x+1, y);
                GridAgent rightNeighbour =
                grid.getAgent(x, (y >= grid.getY()-1)?0:y+1);

          // right neighbour
          double rv =
          sAgent.getSimilarityToNeighbours().get(rightNeighbour);


          // bottom neighbour
          double dv =
          sAgent.getSimilarityToNeighbours().get(downNeighbour);
          DecimalFormat df = new DecimalFormat("#.##");
          df.setMinimumFractionDigits(2);
          r1 += "\u265F \u2014 " + df.format(rv) + " \u2014 ";
          r2 += "|          ";
          r3 += ""  + df.format(dv) + "       ";
                }
                System.out.println(r1);
                System.out.println(r2);
                System.out.println(r3);
                System.out.println(r2);
                }
System.out.println("Performance: " +grid.getCurrentAvgIndKnowledge());
System.out.println("------------------------------------------------------------
------------------------------------------------");
System.out.println();
                }
          }
if (t == pointOfSignificantChange + afterPointOfSignificantChange) {

    // Counting groups
    List<Agent> sortedAgents = new ArrayList<Agent>(
    grid.getAgentsList());
    Collections.sort(sortedAgents);
    int groups = 1;
    int[][] groupGrid = new int[gridSize][gridSize];
```

```java
        for (int i = 0; i < sortedAgents.size() - 1; i++) {
                GridAgent cAgent = (GridAgent) sortedAgents.get(i);
                groupGrid[cAgent.getX()][cAgent.getY()] = groups;
                if (sortedAgents.get(i).compareTo(
                        sortedAgents.get(i + 1)) != 0) {
                                groups++;
                                        }
                                }
        groupsAfterShock.add(groups);
        GridAgent cAgent = (GridAgent)
        sortedAgents.get(sortedAgents.size() -1);

        groupGrid[cAgent.getX()][cAgent.getY()] = groups;

// Print visualization of grid and group membership
for(int i = 0; i < groupGrid.length; i++){
        for (int j = 0; j < groupGrid[i].length; j++) {
                System.out.print(groupGrid[i][j] + " ");
                        }
                System.out.println();
                }
                System.out.println();
                System.out.println();
                        if(sameAgentsForEveryIteration){
                                for(int x = 0; x < grid.getX(); x++){
                                        String r1 = "";
                                        String r3 = "";
                                        String r2 = "";
                                        for(int y = 0; y < grid.getY(); y++){
                                        SimilarityAgent sAgent =
                                        (SimilarityAgent) grid.getAgent(x, y);
sAgent.calculateSimilarityToNeighbours();
GridAgent downNeighbour =
grid.getAgent((x >= grid.getX()-1)?0:x+1, y);

GridAgent rightNeighbour =
grid.getAgent(x, (y >= grid.getY()-1)?0:y+1);

// right neighbour
double rv =
sAgent.getAccordanceValue(rightNeighbour.getKnowledge())/75.0;

// bottom neighbour
double dv = sAgent.getSimilarityToNeighbours().get(downNeighbour);
df.setMinimumFractionDigits(2);
        r1 += "\u265F \u2014 " + df.format(rv) + " \u2014 ";
        r2 += "|           ";
        r3 += ""  + df.format(dv) + "        ";
                }
                System.out.println(r1);
                System.out.println(r2);
```

```
            System.out.println(r3);
            System.out.println(r2);
            }

      System.out.println("Performance: " +
      grid.getCurrentAvgIndKnowledge());
                        }
                  }

if (t == pointOfSignificantChange) {
      grid.turbulenceReality(2, 30, 1);
                        }
      if (t == pointOfSignificantChange) {
            if(turnover)
            grid.turnoverAgents();
            else{
            int agentsToSwap =
            (int) (turnoverRate * grid.getX() * grid.getY());
            int agentsSwapped = grid.turnOverInGroups(turnoverType,
      agentsToSwap, singleRun);
                  if (agentsToSwap != agentsSwapped) {
                  System.out.println("Only "+ agentsSwapped + "instead
            of " + agentsToSwap + " agents were changed.");
                        }
                  }
            grid.rotateAgents();
                  }
            } else {

// More than one iteration
if (t == pointOfSignificantChange-beforePointOfSignificantChange){

// counting groups
List<Agent> sortedAgents = new ArrayList<Agent>(
grid.getAgentsList());
Collections.sort(sortedAgents);
int groups = 1;
int[][] groupGrid = new int[gridSize][gridSize];
      for (int i = 0; i < sortedAgents.size() - 1; i++) {
            GridAgent cAgent = (GridAgent) sortedAgents.get(i);
            groupGrid[cAgent.getX()][cAgent.getY()] = groups;
            if (sortedAgents.get(i).compareTo(sortedAgents.get(i +
      1)) != 0) {
                  groups++;
                        }
                  }
            groupsBeforeShock.add(groups);
            }
                  if (t == pointOfSignificantChange) {
                  grid.turbulenceReality(2, 30, 1);
                        if (turnover)
```

```
                        grid.turnoverAgents();
                        else {
                        int agentsToSwap = (int) (turnoverRate
                        * grid.getX() * grid.getY());
                        int agentsSwapped =
                        grid.turnOverInGroups(turnoverType,
                        agentsToSwap, singleRun);
                              if (actualSwaps.get(agentsSwapped) ==
                        null) {
                              actualSwaps.put(agentsSwapped, 1);
                              } else {
                              actualSwaps.put(agentsSwapped,
                              actualSwaps.get(agentsSwapped) + 1);
                                    }
                              }
                        grid.rotateAgents();
                              }
                              if (t == pointOfSignificantChange +
                              afterPointOfSignificantChange){
                              // Count groups
                              List<Agent> sortedAgents =
                              new ArrayList<Agent>(
                              grid.getAgentsList());
                              Collections.sort(sortedAgents);
                              int groups = 1;
                              int[][] groupGrid =
                              new int[gridSize][gridSize];
                              for (int i = 0; i < sortedAgents.size() -
                              1; i++) {
                                    GridAgent cAgent = (GridAgent)
                                    sortedAgents.get(i);
                                    groupGrid[cAgent.getX()]
                                    [cAgent.getY()] = groups;
                              if (sortedAgents.get(i).compareTo(
                                    sortedAgents.get(i + 1)) != 0) {
                                          groups++;
                                          }
                                    }
                                    groupsAfterShock.add(groups);
                              }
if (t > pointOfSignificantChange && (currentKnowledge >= threshhold)) {

      // Organization runlocked organizational path
      recovered = true;
                  }
            }
      }// end of ticks

// Performance measurement
if(showLastPerformanceValues){
 lastPerformanceValues.add(grid.getCurrentAvgIndKnowledge());
```

```
    }
    if(recovered){
        recoveredCounter++;
            }

        // Counting groups
        List<Agent> sortedAgents = new ArrayList<Agent>(
            grid.getAgentsList());
            Collections.sort(sortedAgents);
            int groups = 1;
            for (int i = 0; i < sortedAgents.size() - 1; i++) {
                if (sortedAgents.get(i).compareTo
            (sortedAgents.get(i + 1)) != 0) {
                    groups++;
                }
            }
            groupsEndOfTicks.add(groups);
}// end of iterations

if(!sameAgentsForEveryIteration) {
    for (int t = 0; t < steps; t++) {
        System.out.println("t= " + t + ": " +
        sumAvgIndKnowledge.get(t) / iterations);
            }
        }
        int summe = 0;
        int groupsDiffer = 0;
        for(int i = 0; i < iterations; i++){

    if(groupsBeforeShock.get(i) !=(groupsAfterShock.get(i))){
                groupsDiffer++;
            }
        }
        System.out.println(groupsDiffer + "/" + iterations
        + "Number of changes in group + (" +
        df.format(groupsDiffer * 1.0 / iterations) + ")");

        for (int gruppen : groupsEndOfTicks) {
            summe += gruppen;
        }
        System.out.println("Average number of groups at the end
    of iteration: " + ((double) summe) / iterations);

        System.out.println("Of " + recoveredCounter + " from " +
    iterations + " runs unlocking occurs.");

        if (showLastPerformanceValues) {
            Collections.sort(lastPerformanceValues);
            for(int i = 0; i < lastPerformanceValues.size();
        i++){
                System.out.println(i + ": " +
```

```java
                    lastPerformanceValues.get(i));
            }
        }

        double average = 0;
        for (int i = 0; i < iterations; i++) {
            average += learningInteractionChanged.get(i);
        }
        average /= iterations * grid.getAgentsList().size();
        System.out.println("Percentage of agents, that changed
    the learning buddy after shock: " + average);
    }

private static List<Agent> formTMT(double bestPercentage) {

List<Agent> tmt = new ArrayList<Agent>(grid.getAgentsList());
final Map<Agent, Integer> tmtMap = new HashMap<Agent, Integer>();
    for (Agent a : tmt) {

    // calculate accordance values for comparing/sorting
    tmtMap.put(a, a.getAccordanceValue(grid.getReality()));
        }

    //Sort according to similarity to reality (descending order!)
    Collections.sort(tmt, new Comparator<Agent>() {
        @Override
        public int compare(Agent o1, Agent o2) {
            if (tmtMap.get(o1) < tmtMap.get(o2)) {
                    return 1;
                }
                if (tmtMap.get(o1) > tmtMap.get(o2)) {
                    return -1;
                }
                    return 0;
            }
        });
        int n = (int) Math.round(tmt.size() * bestPercentage);

        //Take n best agents for TMT
        return tmt.subList(0, n);
    }

private static ArrayList<GridAgent> rumbleList(List<Agent> agents) {
    Random rnd = new Random();
    List<Agent> l = new ArrayList<Agent>(agents);
    ArrayList<GridAgent> result = new
    ArrayList<GridAgent>(agents.size());
    int size = l.size();
        for (int i = 0; i < size; i++) {
            int index = rnd.nextInt(l.size());
            result.add((GridAgent) l.remove(index));
```

```
        }
        return result;
    }

private static ArrayList<Integer> generateKnowledge(int dimensions){
    ArrayList<Integer> result = new    ArrayList<Integer>(dimensions);
        Random rnd = new Random();
        for (int j = 0; j < dimensions; j++) {
            double p = rnd.nextDouble();
            int value;
            if (p < 1.0 / 3.0) {
                value = -1;
            } else if (p < 2.0 / 3.0) {
                value = 1;
            } else {
                value = 0;
            }

            result.add(value);
        }
        return result;
    }

    private static Grid grid;
    private static DecimalFormat df = new DecimalFormat("#.##");
}
//END
```

A.2.2 Agent Logic ("SimilarityAgent.java")

```
/*
 *  SimilarityAgent.java
 *
 *  (c) 2013, Felix Obschonka
 *
 */

package learningAgents.gridNetworking;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Random;
import learningAgents.Agent;

//Defines learning of agent according to similarity algorithm
public class SimilarityAgent extends GridAgent {

	private boolean isPartOfTMT;
	private HashMap<SimilarityAgent, Double> similarityMap =
	new HashMap<SimilarityAgent, Double>();
	private SimilarityAgent currentLearningBuddy = null;
	public SimilarityAgent(int dimensions, int posX, int posY,Grid g)
{
		super(dimensions, posX, posY, g, 0, false);
	}

	//Learning from best neighbour
	public void learnFromBestNeighbour(double x, double y) {
		assert (x + y == 1);
		ArrayList<Integer> reality = this.getGrid().getReality();
		List<GridAgent> neighbors =
		rumbleList(getDirectNeighbours());
		SimilarityAgent buddy = null;
		double value = this.getAccordanceValue(reality);
		for (GridAgent neighbour : neighbors) {
			double v = x *
			this.getAccordanceValue(neighbour.getKnowledge())
			+ y * neighbour.getAccordanceValue(reality);
				if (v > value) {
					value = v;
					buddy = (SimilarityAgent) neighbour;
					break;
			}
		}
		if (buddy != null)
```

```
                super.learn(buddy, similarity(buddy)
                            / (double) getKnowledge().size());
            this.currentLearningBuddy = buddy;
      }
      private int similarity(Agent that) {
            int result = 0;
            for (int i = 0; i < this.getKnowledge().size(); i++) {
                  result += (this.getKnowledge().get(i) ==
                  that.getKnowledge().get(i)) ? 1 : 0;
            }
            return result;
      }
      public void promoteToTMT(){
            isPartOfTMT = true;
      }
      public boolean isTMTMember(){
            return isPartOfTMT;
      }
      public HashMap<SimilarityAgent, Double>
      getSimilarityToNeighbours(){
            return this.similarityMap;
      }
      public void calculateSimilarityToNeighbours(){
            List<GridAgent> neighbors = getDirectNeighbours();
            for(GridAgent neighbour : neighbors){
            similarityMap.put((SimilarityAgent) neighbour,
            (double)similarity(neighbour) /
            this.getKnowledge().size());
            }
      }
      private static ArrayList<GridAgent>
      rumbleList(List<GridAgent> agents) {
            Random rnd = new Random();
            List<GridAgent> l = new ArrayList<GridAgent>(agents);
            ArrayList<GridAgent> result = new
            ArrayList<GridAgent>(agents.size());
            int size = l.size();
            for (int i = 0; i < size; i++) {
                  int index = rnd.nextInt(l.size());
                  result.add(l.remove(index));
            }
            return result;
      }
      public SimilarityAgent getCurrentLearningBuddy() {
            return currentLearningBuddy;
      }
}
//END
```

A.2.3 Grid ("Grid.java")

```java
/*
 * Grid.java
 *
 * (c) 2013, Felix Obschonka
 *
 */
package learningAgents.gridNetworking;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;
import java.util.Vector;
import learningAgents.Agent;
import learningAgents.Environment;

public class Grid extends Environment {

//Initialization & Constructor
private int x, y;
private Vector<Vector<GridAgent>> agents;
private double turbulenceProbability;
private double turnoverProbability;
private double rotateProbability;
private List<Agent> agentList = null;
protected boolean generateNewAgentsList = true;
private ArrayList<Integer> usedDimensiones = new ArrayList<Integer>();

	public Vector<Vector<GridAgent>> getAgents() {
		return agents;
	}

	public Grid(int x, int y, int sizeOfAgents,
			double turbulenceProbability, double
		turnoverProbability, double rotateProbability,
			double alpha) {
		super(sizeOfAgents);
		this.x = x;
		this.y = y;
		this.turbulenceProbability = turbulenceProbability;
		this.rotateProbability = rotateProbability;
		this.turnoverProbability = turnoverProbability;
		generateAgents();
	}

	public Grid(int x, int sizeOfAgents,
			double turbulenceProbability, double
		turnoverProbability, double rotateProbability,
```

```
            double alpha) {
        this(x, x, sizeOfAgents, turbulenceProbability,
                    turnoverProbability, rotateProbability,
                     alpha);
}

//Generates agents in the grid
private void generateAgents() {
        this.agents = new Vector<Vector<GridAgent>>();
        for (int i = 0; i < x; i++) {
            Vector<GridAgent> row = new Vector<GridAgent>(y);
            for (int j = 0; j < y; j++) {
            row.add(new SimilarityAgent(sizeOfAgents,i, j, this));
            }
            this.agents.add(row);
        }
}
@Override
public double getCurrentAvgIndKnowledge() {
        double result = 0.0;
        for (Vector<GridAgent> row : this.agents) {
            for (GridAgent agent : row) {
            int v = agent.getAccordanceValue(reality);
            result += (double) v / ((double) sizeOfAgents);
            }
        }
        return result / (x * y);
}

public GridAgent getAgent(int x, int y) {
return agents.get(x).get(y);
}

public int getX() {
        return x;
}

public int getY() {
        return y;
}
/*
 * Defines the environmental change:
 *  1. Incremental change
 *  2. Environmental shock
 *  3. Change in opposite direction
 *  4. Change on defined number of dimensions
 */
public void turbulenceReality(int method, int dimensions,
int turbulenceRange) {
        Random rnd = new Random();
        switch (method) {
```

```
case 1: // incremental
      for (int i = 0; i < sizeOfAgents; i++) {
            if (rnd.nextDouble() < turbulenceProbability)
      {
            int value = ((int) rnd.nextInt(2)) * 2 - 1;
                  reality.set(i, value);
            }
      }
      return;
case 2: { // shock
      double currentProb = rnd.nextDouble();
            if (currentProb <= turbulenceProbability) {
            reality = generateReality();
      }
      return;
}
case 3: // change reality to the opposite
      double currentProb = rnd.nextDouble();
            if (currentProb <= turbulenceProbability) {
            for (int i = 0; i < reality.size(); i++) {
                  if (reality.get(i) == 1) {
                        reality.set(i, -1);
                  } else {
                        reality.set(i, 1);
                  }
            }
      }
      return;
case 4:{ // change on a number of dimensions
      double c = (double)dimensions /
      ((double)turbulenceRange);
      int count = 0;
      if (c == Math.floor(c)){
            count = (int) c;
      } else {
            if(rnd.nextDouble() < (c-Math.floor(c))){
                  count = (int) Math.ceil(c);
            } else {
                  count = (int) Math.floor(c);
            }
      }
      ArrayList<Integer> dimensionsToChange = new
      ArrayList<Integer>(count);
      for(int i = 0; i < count; i++){
            int a;
            do {
                  a = rnd.nextInt(this.sizeOfAgents);
            } while (dimensionsToChange.contains(a) ||
            usedDimensiones.contains(a));
            dimensionsToChange.add(a);
      }
```

```
            for(int dimension : dimensionsToChange){
                usedDimensiones.add(dimension);
                if(reality.get(dimension) > 0){
                    reality.set(dimension, -1);
                } else {
                    reality.set(dimension, 1);
                }
            }
            return;
            }
        default:
            throw new RuntimeException("Change: Select 1-4!");
        }
}
// Turnover of agents in the grid
@Override
public void turnoverAgents() {
    Random rnd = new Random();
    for (int i = 0; i < agents.size(); i++) {
    for (int j = 0; j < agents.get(i).size(); j++) {
        if (rnd.nextDouble() < turnoverProbability) {
            GridAgent oldAgent = agents.get(i).get(j);
            for (GridAgent a : oldAgent.getTeacherTo())
                a.getLearningBuddies().remove(oldAgent);
                SimilarityAgent a = (SimilarityAgent)
                agents.get(i).get(j);
                ArrayList<Integer> newKnowledge = new
                ArrayList<Integer>();
                if (!a.isTMTMember()) {
                    for (int d = 0; d < sizeOfAgents;
                    d++) {
                    newKnowledge.add(rnd.nextInt(3)- 1);
                    }
                    a.setKnowledge(newKnowledge);
                }
            }
        }
    }
}
/* For SimilarityAgents only:
 * Look up groups and change agents
 * on edges (if onEdge true) or in
 * the center of a group (otherwise)
 */
public int turnOverInGroups(boolean onEdge, int
agentsToSwitch, boolean printSwap) {
    int c = 0;
    for(Agent a1 : this.getAgentsList()){
        SimilarityAgent sAgent = (SimilarityAgent) a1;
        sAgent.calculateSimilarityToNeighbours();
        boolean fits;
```

```
        if(onEdge){
              fits = false;
              for(double v :
              sAgent.getSimilarityToNeighbours().values()){
                    fits |= v != 1;
              }
        } else {
              fits = true;
              for(double v :
              sAgent.getSimilarityToNeighbours().values()){
                    fits &= v == 1;
              }
        }
        if(fits){
              c++;
        }
        if(c >= agentsToSwitch)
              break;
  }
  if(c < agentsToSwitch){
        agentsToSwitch = c;
  }
  ArrayList<String> taken = new ArrayList<String>();
  List<Agent> agents = this.getAgentsList();
  List<SimilarityAgent> toSwap = new
        ArrayList<SimilarityAgent>(agentsToSwitch);
  Random rnd = new Random();
  for(int i = 0; i < agentsToSwitch; i++){
        int r = -1;
        boolean fits;
        SimilarityAgent sAgent = null;
        do{
              r = rnd.nextInt(agents.size());
              sAgent = (SimilarityAgent) agents.get(r);
              if(onEdge){
                    fits = false;
                    for(double v :
              sAgent.getSimilarityToNeighbours().values()){
                          fits |= v != 1;
                    }
              } else {
                    fits = true;
                    for(double v :
        sAgent.getSimilarityToNeighbours().values()){
                          fits &= v == 1;
                    }
              }
        }while(!fits || taken.contains(sAgent.getX() + "" +
  getY()));
        taken.add(sAgent.getX() + "" + sAgent.getY());
        toSwap.add(sAgent);
```

```
        }
        for(SimilarityAgent agent : toSwap){
            ArrayList<Integer> knowledge = new
            ArrayList<Integer>(this.sizeOfAgents);
            for (int j = 0; j < this.sizeOfAgents; j++) {
                double p = rnd.nextDouble();
                int value = (int) Math.floor(p * 3) - 1;
                knowledge.add(value);
            }
            agent.setKnowledge(knowledge);
                if(printSwap){
                System.out.println("agent swapped at x: " +
                agent.getX() + " y: " + agent.getY());
            }
        }
        return toSwap.size();
    }
    //Method to rotate agents in the grid
    public void rotateAgents(){
        if (rotateProbability <= 0.0)
            return;
        Random rnd = new Random();
        for (int i = 0; i < agents.size(); i++) {
            for (int j = 0; j < agents.get(i).size(); j++) {
                if (rnd.nextDouble() < rotateProbability) {
                    int x, y;
                    do {
                    x = rnd.nextInt(agents.size());
                    y = rnd.nextInt(agents.get(i).size());
                    } while (x == i && y == j);
                    GridAgent swapAgent =
                    agents.get(x).get(y);
                    GridAgent currentAgent =
                    agents.get(i).get(j);
                    ArrayList<Integer> helpKnow =
                    swapAgent.getKnowledge();
            swapAgent.setKnowledge(currentAgent.getKnowledge());
                    currentAgent.setKnowledge(helpKnow);
                }
            }
        }
    }
    @Override
    public List<Agent> getAgentsList() {
        if (this.agentList == null || generateNewAgentsList) {
            agentList = new ArrayList<Agent>(this.x * this.y);
            for (Vector<GridAgent> row : this.agents) {
                for (GridAgent agent : row) {
                    agentList.add(agent);
                }
            }
```

```
                generateNewAgentsList = false;
            }
            return agentList;
        }
        //Reset dimensions for next iteration
        public void resetUsedDimensions(){
            this.usedDimensiones = new ArrayList<Integer>();
        }
        public void setAgents(Vector<Vector<GridAgent>> agents){
            this.agents = agents;
        }
}//END
```

A.2.4 Agent ("Agent.java")

```
/*
 *  Agent.java
 *
 *  (c) 2013, Felix Obschonka
 *
 */
package learningAgents;
import java.util.Random;
import java.util.ArrayList;

public abstract class Agent implements Comparable<Agent>{

	protected ArrayList<Integer> knowledge;

	public Agent(int dimensions) {
		knowledge = new ArrayList<Integer>(dimensions);
		Random rnd = new Random();
		for (int j = 0; j < dimensions; j++) {
			double p = rnd.nextDouble();
			int value;
				if (p < 1.0 / 3.0) {
					value = -1;
				} else if (p < 2.0 / 3.0) {
					value = 1;
				} else {
					value = 0;
			}
			knowledge.add(value);
		}
	}

	public void init(){
		int dimensions = knowledge.size();
		knowledge = new ArrayList<Integer>(dimensions);
		Random rnd = new Random();
		for (int j = 0; j < dimensions; j++) {
			double p = rnd.nextDouble();
			int value;
			if (p < 1.0 / 3.0) {
				value = -1;
			} else if (p < 2.0 / 3.0) {
				value = 1;
			} else {
				value = 0;
			}
			knowledge.add(value);
		}
```

```
	}

	public int getAccordanceValue(ArrayList<Integer> reality) {
		assert (reality.size() == knowledge.size());
		int result = 0;
		for (int i = 0; i < reality.size(); i++) {
			result += reality.get(i) * this.knowledge.get(i);
		}
		return result;
	}

	public ArrayList<Integer> getKnowledge() {
		return knowledge;
	}

	public void setKnowledge(ArrayList<Integer> knowledge) {
		this.knowledge = knowledge;
	}

	public void setKnowledge(int i, int v) {
		this.knowledge.set(i, v);
	}

	public abstract void learn(Agent agent, double
	learningProbability);

	public abstract void addLearningBuddy(Agent agent);

	@Override
	public int compareTo(Agent that) {
		int dimensions = this.getKnowledge().size();
		for(int i = 0; i < dimensions; i++){
			if(this.getKnowledge().get(i) >
			that.getKnowledge().get(i)){
				return 1;
			}
			if(this.getKnowledge().get(i) <
			that.getKnowledge().get(i)){
				return -1;
			}
		}
		return 0;
	}
}//END
```

A.2.5 Agent Behavior ("GridAgent.java")

```
/*
 *  GridAgent.java
 *
 *  (c) 2013, Felix Obschonka
 *
 */
package learningAgents.gridNetworking;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Random;
import java.util.Vector;
import learningAgents.Agent;

public class GridAgent extends Agent {

	// Initialization
	private int x, y;
	private Grid grid;
	private Vector<GridAgent> learningFrom = new Vector<GridAgent>();
	private Vector<GridAgent> teacherTo = new Vector<GridAgent>();
	private boolean learningBuddiesFix;
	private int learningBuddies;
	protected ArrayList<GridAgent> allNeighbours;
	protected boolean generateNewNeighboursList;
	public GridAgent(int dimensions, int x, int y, Grid g, int
	learningBuddies, boolean learningBuddiesFix) {
		super(dimensions);
		this.x = x;
		this.y = y;
		this.grid = g;
		this.teacherTo = new Vector<GridAgent>();
		this.learningFrom = new
		Vector<GridAgent>(learningBuddies);
		this.learningBuddiesFix = learningBuddiesFix;
		this.learningBuddies = learningBuddies;
	}
	public Vector<GridAgent> getLearningBuddies() {
		return learningFrom;
	}
	@Override
	public void addLearningBuddy(Agent agent) {
		GridAgent a = (GridAgent) agent;
		this.learningFrom.add(a);
		a.teacherTo.add(this);
```

```
}

public void addTeacherTo(GridAgent teacherTo) {
      this.teacherTo.add(teacherTo);
}
public int getX() {
      return x;
}
public int getY() {
      return y;
}

//Creates list of all neighbors
public List<GridAgent> getAllNeighbours() {
      if (this.allNeighbours == null ||
      this.generateNewNeighboursList) {
      ArrayList<GridAgent> result = new  ArrayList<GridAgent>(8);
            int a, b;

            a = (x > 0) ? x - 1 : grid.getX() - 1;
            b = y;
            result.add(grid.getAgent(a, b));

            a = (x < grid.getX() - 1) ? x + 1 : 0;
            b = y;
            result.add(grid.getAgent(a, b));

            a = x;
            b = (y < grid.getY() - 1) ? y + 1 : 0;
            result.add(grid.getAgent(a, b));

            a = x;
            b = (y > 0) ? y - 1 : grid.getY() - 1;
            result.add(grid.getAgent(a, b));

            a = (x > 0) ? x - 1 : grid.getX() - 1;
            b = (y < grid.getY() - 1) ? y + 1 : 0;
            result.add(grid.getAgent(a, b));

            a = (x > 0) ? x - 1 : grid.getX() - 1;
            b = (y > 0) ? y - 1 : grid.getY() - 1;
            result.add(grid.getAgent(a, b));

            a = (x < grid.getX() - 1) ? x + 1 : 0;
            b = (y < grid.getY() - 1) ? y + 1 : 0;
            result.add(grid.getAgent(a, b));

            a = (x < grid.getX() - 1) ? x + 1 : 0;
            b = (y > 0) ? y - 1 : grid.getY() - 1;
            result.add(grid.getAgent(a, b));
```

```
            this.allNeighbours = result;
        }
        return this.allNeighbours;
    }

    //Randomly selects a neighbour
    public List<GridAgent> getSomeRandomNeighbours(int number) {
        List<GridAgent> result = getAllNeighbours();
        Random rnd = new Random();
        int size = result.size();
        for (int i = number; i < size; i++) {
            result.remove(rnd.nextInt(result.size()));
        }
        return result;
    }

    //Initiates learning process
    @Override
    public void learn(Agent agent, double learningProbability) {
        Random rnd = new Random();
        for (int i = 0; i < this.knowledge.size(); i++) {
            if (rnd.nextDouble() < learningProbability) {
                if (this.knowledge.get(i) *
                agent.getKnowledge().get(i) == -1) {
                    this.knowledge.set(i, 0);
                } else if (this.knowledge.get(i) == 0) {
                    this.knowledge.set(i,
                agent.getKnowledge().get(i));
                }
            }
        }
    }

    public Vector<GridAgent> getTeacherTo() {
        return teacherTo;
    }

    public void resetTeaching() {
        this.teacherTo = new Vector<GridAgent>();
    }

    public Grid getGrid() {
        return grid;
    }

    public void resetLearningBuddies() {
        if(learningFrom != null)
        for (GridAgent agent : learningFrom) {
            agent.teacherTo.remove(agent);
        }
```

```
        if(teacherTo != null)
        for (GridAgent agent : teacherTo) {
              agent.learningFrom.remove(agent);
        }
        this.learningFrom = null;
        this.teacherTo = null;
  }

  public void setX(int x) {
        this.x = x;
  }

  public void setY(int y) {
        this.y = y;
  }

  public void defineLearningBuddies() {

  }

  public List<GridAgent> getDirectNeighbours(){
        List<GridAgent> a = getAllNeighbours();
        return a.subList(0,4);
  }
}
//END
```

A.2.6 Environment ("Environment.java")

```java
/*
 *  Environment.java
 *
 *  (c) 2013, Felix Obschonka
 *
 */
package learningAgents;
import java.util.List;
import java.util.Random;
import java.util.ArrayList;

public abstract class Environment {
	protected ArrayList<Integer> reality;
	protected int sizeOfAgents;
	public Environment(int sizeOfAgents) {
		this.sizeOfAgents = sizeOfAgents;
		this.reality = generateReality();
	}

	//Randomly generates an environment
	protected ArrayList<Integer> generateReality() {
		ArrayList<Integer> result = new
		ArrayList<Integer>(sizeOfAgents);
		Random rnd = new Random();
		for (int i = 0; i < sizeOfAgents; i++) {
			double p = rnd.nextDouble();
			int value = ((int) Math.round(p)) * 2 - 1;
			result.add(value);
		}
		return result;
	}
	public ArrayList<Integer> getReality() {
		return reality;
	}
	public abstract double getCurrentAvgIndKnowledge();

	//Exchange agents according to reality
	public abstract void turnoverAgents();
	public abstract void rotateAgents();
	public abstract List<Agent> getAgentsList();

	//Changes reality according to the parameters set in main
	public abstract void turbulenceReality(int method, int
	dimensions, int turbulenceRange);
} //END
```

A.3 Publications

Obschonka, F.; Petermann, A. (2013). The Influence Of Management For Breaking Organizational Paths - A Simulation Study, ECMS 2013 Proceedings edited by: W. Rekdalsbakken, R. T. Bye, H. Zhang, European Council for Modeling and Simulation. doi:10.7148/2013-0322