An Intersubjective Perspective on Entrepreneurial Decision Making
– a Computer Simulation Study on the Implications of Networks and Interactions for Effectuation

Von der Fakultät für Wirtschaftswissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Wirtschafts- und Sozialwissenschaften genehmigte Dissertation

vorgelegt von
Dipl.-Ing. Jan Willem Jansen

Berichter: Univ.-Prof. Dr.rer.pol.habil. Malte Brettel
Univ.-Prof. Dr.rer.pol. Frank Thomas Piller

Tag der mündlichen Prüfung: 03. Juni 2015

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar
For Helen
Both practitioners and scientists regard entrepreneurship as an important driver of product and market innovation. Researchers have identified numerous strategies employed by entrepreneurs to successfully introduce new products and services in new markets. Most strategies employ prediction of future market characteristics as a key ingredient. “The business plan” has become a ubiquitous buzzword for entrepreneurial practitioners and scientists alike.

In 2001, effectuation was presented as a radically different approach to entrepreneurship designed especially for highly uncertain environments. Derived from the study of expert serial entrepreneurs, effectuation focuses on controlling market outcomes rather than predicting them. Effectuation is a decision logic that – due to high uncertainty – rejects predictions about the future. Consequently, it favours working with available means over setting potentially unrealistic goals. Moreover, effectuation proposes to focus on the potential loss of a project rather than its expected returns. In addition, effectuation encourages seeing other market participants as potential partners, not as potential threats. Lastly, effectuation regards unforeseen contingencies as improvement opportunities rather than incidents to be avoided.

Despite significant research attention, effectuation is still in a nascent state: processes are not yet fully understood, the discussion regarding guiding principles is still on-going. Bigger challenges still lay ahead: leading entrepreneurship researchers aim to transform entrepreneurship into a “science of the artificial” and recreate it as a set of methodologies that can be tailored and optimised for individual entrepreneurial scenarios. In order to qualify as a suitable approach, however, a deeper understanding of effectuation mechanisms is required.
A key component of effectuation is intersubjective\textsuperscript{1} interaction. The reliance on co-creation and the inclusive attitude towards other market participants distinguishes effectuation from other entrepreneurial strategies. However, its processes, consequences, and circumstances were also identified as a key area in need for further research attention in effectuation literature.

Consequently, this thesis investigates the implications of networks and interactions for effectuation from an intersubjective perspective. It reveals the importance of docile and persistent market participants, challenges classic entrepreneurial assumptions regarding network position and shape, and reveals docility\textsuperscript{2} as a precursor for successful entrepreneurship in highly complex environments.

With computer simulation, this thesis uses a methodology that is starkly underrepresented in entrepreneurship research and in social sciences in general. Computer simulation fits the task of understanding complex, non-linear relationships between multitudes of individuals very well. Computer simulation allows the longitudinal study of a large number of individuals at high level of detail – far beyond empirical feasibility. Moreover, it allows testing current theories under boundary conditions by performing “virtual experiments” which contributes greatly to a better understanding of theory.

Computer simulation requires a formal model of the subject under investigation. The creation of a model of effectuation combines available knowledge regarding principles and processes. It thus fosters more precise terminology and rigour in theory development. Hence, the formal model is an important theoretical contribution in itself. While results derived from computer simulation results require – like theoretical predictions – empirical validation, the methodology facilitates empirical research by providing well-grounded hypotheses.

\textsuperscript{1} intersubjective: „involving or occurring between separate conscious minds“ (Merriam Webster)
\textsuperscript{2} docility: teachability or educatability
Overview of papers

The following research papers (or earlier versions thereof) were accepted and presented at scientific research conferences. Part B of this thesis contains the latest version of each.

Paper I – Individual vs. Collective Control in Effectual Social Networking: A Simulation Study


Table of Contents

Summary........................................................................................................................................ V

Part A. Introductory overview of the thesis .................................................................................. X

1 Introduction.................................................................................................................................. 1
  1.1 Motivation and relevance........................................................................................................ 3
  1.2 Research framework and research questions ..................................................................... 5
  1.3 Aspired contribution ............................................................................................................. 12

2 Theoretical and conceptual background .................................................................................. 15
  2.1 Effectuation........................................................................................................................... 15
     2.1.1 Introduction and classification....................................................................................... 15
     2.1.2 Key principles ............................................................................................................. 17
     2.1.3 Effectual problem space ............................................................................................ 20
     2.1.4 Effectuation from a process perspective .................................................................... 21
     2.1.5 Current state of effectuation research ........................................................................ 24
  2.2 Social networks....................................................................................................................... 27
     2.2.1 Analysis of network position ...................................................................................... 29
     2.2.2 Analysis of network shape ........................................................................................... 30

3 Research methodology.............................................................................................................. 33
  3.1 Introduction to computer simulation ................................................................................... 33
  3.2 Methodological fit for entrepreneurship research ............................................................. 36
  3.3 Simulation creation process and nature of outcomes ......................................................... 38

4 Summary of research papers.................................................................................................... 41
  4.1 Paper I: Individual vs. Collective Control in Effectual Social Networking: A Simulation Study ...................................................................................................................... 42
  4.3 Paper III: Out of Thin Air? Simulating Entrepreneurial Opportunity Creation and the Impact of Environmental Complexity and Stakeholder Behavior ......................................................... 47
5 Overall implications

5.1 Theoretical implications

5.2 Areas of further research

5.3 Managerial implications

References

Appendices

Appendix A – Overview of simulator structure

Appendix B – Simulation source code

Part B. Research papers

Individual vs. Collective Control in Effectual Social Networking: A Simulation Study

Entrepreneurial Mingling Secrets: Investigating the Performance Impact of Network Structure for Control-Based Entrepreneurship using Agent-based Simulation

Out of Thin Air? Simulating Entrepreneurial Opportunity Creation and the Impact of Environmental Complexity and Stakeholder Behavior
Part A. Introductory overview of the thesis

Table of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AoM</td>
<td>Academy of Management</td>
</tr>
<tr>
<td>DS</td>
<td>Demand satisfaction</td>
</tr>
<tr>
<td>e.g.</td>
<td>Exempli gratia = for example</td>
</tr>
<tr>
<td>EURAM</td>
<td>European Academy of Management</td>
</tr>
<tr>
<td>i.e.</td>
<td>Id est = that is</td>
</tr>
<tr>
<td>p.</td>
<td>Page</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Research and development</td>
</tr>
<tr>
<td>RWTH</td>
<td>Rheinisch-Westfälische Technische Hochschule</td>
</tr>
</tbody>
</table>

Table of figures

- Figure 1: Overview of research framework .......................................................... 6
- Figure 2: Control-Prediction matrix ................................................................. 16
- Figure 3: Causation vs. effectuation in market creation ...................................... 20
- Figure 4: The two dynamic cycles of effectuation ............................................... 22
- Figure 5: The behavioural theory of the entrepreneurial firm ............................. 23
- Figure 6: Interaction of computer simulation and theory development ............ 39

Table of tables

- Table 1: Contribution to literature by paper ....................................................... 13
- Table 2: Principles of effectuation and causation ............................................... 19
- Table 3: Effectual transformation mechanisms .................................................... 24
- Table 4: Requirements for the application of computer simulation .................... 38
1 Introduction

Entrepreneurship, the “process of creative destruction” (Schumpeter, 1942), is widely recognised as a key driver behind the creation of radically innovative products and services. While regarded as a destructor of established markets and business models, entrepreneurship is seen as a core driver of improvement and economic welfare. Scientifically introduced by Schumpeter (1934), research on entrepreneurship as a phenomenon and on the entrepreneur as a person has developed into an important area of research in social sciences (Shane & Venkataraman, 2000).

While entrepreneurship research has made tremendous progress since 1934, even bigger challenges still lay ahead: the aspired transformation of entrepreneurship into a “science of the artificial” (Chiles, Gupta, & Bluedorn, 2008; Venkataraman, Sarasvathy, Dew, & Forster, 2012). This means no less than to change the focus of entrepreneurship research from observing “who the entrepreneur is and what he or she does” (Shane & Venkataraman, 2000, p. 218) towards the definition of an “entrepreneurial method” (Sarasvathy & Venkataraman, 2011, p. 125) that can be tailored to specific environmental preconditions and personal circumstances.

One approach that has all the earmarks of becoming an integral part of this science of the artificial is effectuation. Introduced as an alternative entrepreneurial decision logic by Sarasvathy (2001), effectuation is seen as a potential building block by leading entrepreneurship researchers (Venkataraman et al., 2012). Effectuation relies on action-based guidelines, provides a teachable process, incorporates personal preferences, and does not hinge on “personal genius”, but is derived from the decision-making of expert serial entrepreneurs (Sarasvathy & Venkataraman, 2011). It was designed to actively manage the uncertainty usually involved in disruptive innovation. Existing only for little more than a decade, effectuation is still in a nascent research stage (Perry, Chandler, & Markova, 2012 Table 1) and
requires significant further research effort to contribute to the goal of an artificial science of entrepreneurship.

A key area of further research is “making the intersubjective a key unit of analysis” (Sarasvathy & Venkataraman, 2011, p. 125) by exploring the “details of the numerous relationships and deals that entrepreneurs routinely negotiate with a wide variety of stakeholders” (Sarasvathy & Venkataraman, 2011, p. 126). The aim of this thesis is to contribute to this area of research by studying the collective and individual effects of interaction behaviour and the network between effectual entrepreneurs.

Following the pioneering examples of renowned entrepreneurship researchers (Davis, Eisenhardt, & Bingham, 2007; Ganco & Agarwal, 2009; Levinthal, 1997; March, 1991) this thesis uses computer simulation to derive the aforementioned contributions. Given the complex system formed by multiple interacting entrepreneurs, computer simulations “allow scientists to capture much more of the complex causality present in typical entrepreneurial settings” (McKelvey, 2004). While computer simulation as a method is commonplace in engineering and natural sciences, in the social sciences – especially business studies – it has yet to become a prominent method of research. Following in the footsteps of Schlüter (2012), this thesis also aims to promote this method for three reasons. Firstly, it is especially suited for “for advancing theory and research on complex behaviours and systems” (Harrison, Carroll, & Carley, 2007, p. 1). Given the implications of intersubjectivity, entrepreneurship research in general is likely to focus on more complex systems in the future. Secondly, computer simulation requires a high degree of formalization in order to transform concepts and processes into computable algorithms. Making entrepreneurship theory “computable” will therefore reduce ambiguity and foster the clarification of theoretical concepts and constructs. Lastly, the ability to test hypotheses prior to empirical research would allow “traditional” research to build stronger
hypotheses and reduce the effort of the creation (and answering) of surveys as researchers would exactly know what to ask for.

1.1 Motivation and relevance

In her seminal paper, Sarasvathy (Sarasvathy, 2001) introduced effectuation as a radically new entrepreneurial decision logic. Derived from experiments with serial expert entrepreneurs, effectuation was quickly picked up by the scientific community and is now – after only one decade – seen as an important building block for the development of entrepreneurship into a science of the artificial (Venkataraman et al., 2012). Presented as a juxtaposition of causation-based entrepreneurship, it is “relevant to the areas of entrepreneurship research […] because it questions the universal applicability of causation-based models of entrepreneurship” and “represents a paradigmatic shift in the way that we understand entrepreneurship” (Perry et al., 2012, p. 1,2). Moreover, it delivers a convincing explanation for the creation of “future goods and services” in the “absence of current markets” (Perry et al., 2012, p. 21; orig. Venkataraman, 1997).

Despite its high relevance and practical usefulness, effectuation as a scientific theory is still in a nascent state (Perry et al., 2012) and could significantly benefit from further research attention (Chandler, DeTienne, McKelvie, & Mumford, 2011; Perry et al., 2012; Sarasvathy & Venkataraman, 2011).

While Chandler et al. (2011) and Perry et al. (2012) propose further research suitable for theory at an intermediate state regarding “the relationship between effectuation and other established constructs” (2012, p. 12), Sarasvathy and Venkataraman (2011) propose a more fundamental direction of further research. Considering, that after a decade of research the number of basic principles/differences is still not agreed upon (Chandler et al., 2011: at least 3 dimensions; Dew, Read, Sarasvathy, & Wiltbank, 2008: 12 differences; Perry et al., 2012: 5
dimensions; Sarasvathy, 2001: 4 principles), they propose to strengthen the root of effectuation first. Consequently, they propose to make “the [i]ntersubjective a [k]ey [u]nit of [a]nalysis” and explore “how entrepreneurs transform bits and pieces of current realities into valuable new opportunities through productive interaction with others” (both 2011, p. 126). While initial (meta-)studies suggest that the effectual partnering approach has a significantly positive effect on performance (Read, Song, & Smit, 2009), the multi-actor mechanics at work and the effect of individual behaviour or the impact of network shape have not been clarified yet.

A review of available descriptions of the effectuation process (Dew et al., 2008; Sarasvathy & Dew, 2005b) reveals a rather “egocentric” interpretation of intersubjectivity: while “the effectuator” single-handedly manages uncertain situations, the other parties are merely “stakeholders” and assume a rather passive role. This, however, contrasts the idea of “partnering” and “co-creation” as described in Sarasvathy’s initial study (2001). Therefore, a review of the interaction between similarly active individuals seems appropriate. In conclusion, while aspiring to become an important building block of future entrepreneurship theory, effectuation requires significant research effort, especially regarding the impact of interaction including behaviour and networks.

In addition to the relevance regarding the advancement of the effectuation theory, this thesis is also relevant from a methodological point of view. All currently available publications regarding effectuation and the overwhelming majority of studies concerning entrepreneurship in general neglect the utilization of computer simulations. In engineering and natural sciences on the other hand, the use of computer simulation became a commodity decades ago (Harrison et al., 2007). Prototypes of new machinery are stress-tested in simulations prior to manufacturing. Measurements in physical experiments differing from simulation data are usually the point of origin for theory advancements. In the social sciences, dedicated articles that promote the utilization of computer simulation and giving an introduction to the topic are
published in top journals, e.g., *Academy of Management Review* or *Journal of Business Venturing* (Davis et al., 2007; Harrison et al., 2007; McKelvey, 2004). However, examples of actual application of this method are rare, letting the potential of this method lie fallow.

Computer simulation is “a powerful methodology for advancing theory and research on complex behaviours and systems” (Harrison et al., 2007, p. 1). The fully observable nature of computer programs allows for data collection at a level of detail and volume beyond empirical feasibility (Davis et al., 2007). The combination of repeatability and simple adjustment of parameters facilitates the analysis of boundary conditions simply impossible with other research methods. Combining these unique features provides a methodology that enables to go beyond mere dependency statements and to understand why and how an input parameter impacts the output. Taking the nascent state and the complex nature of intersubjective processes into account, computer simulation should provide an effective method to advance the understanding of effectuation theory.

### 1.2 Research framework and research questions

The research deficits are organised in a research framework as presented in Figure 1. Using a computer simulation as core for all research endeavours, this thesis delivers its contributions by using input parameters and measurements on three levels: effectual entrepreneur, network, and opportunity. The simulation core accepts varying interaction behaviour, initial network configurations, and levels of environmental complexity. The simulation allows for measurements on multiple levels at each simulation step enabling longitudinal observations on a large number of actors – a capability unmatched by (most) empirical methods.
Research deficit 1: A formal model of effectual intersubjective behaviour is missing

Sarasvathy, Read, Dew, and Wiltbank gave illustrative overviews of the effectual process (Dew et al., 2008; Dew, Read, Sarasvathy, & Wiltbank, 2010; Sarasvathy & Dew, 2005b) emphasizing different parts such as the iterative character, the inclusion of stakeholders, and the transformation of opportunities. While these overviews exemplified the different aspects of effectuation very well, a unified process model is yet to be created. Effectuation research would benefit from such a unified process model for multiple reasons.

Firstly, a unified model could strengthen theory development by the use of consistent terminology and focus on on-going discussions, e.g. regarding whether opportunities are “made”, “created”, “found”, or “discovered” (Alvarez & Barney, 2013; Shane, 2012; Venkataraman et al., 2012).

Secondly, the proposition to “look […] deeper within simple and direct relationships (signified by arrows in our models) between individual and opportunity” (Venkataraman et al.,
2012, p. 28) or to “Make the Intersubjective a Key Unit of Analysis” (Sarasvathy & Venkataraman, 2011, p. 126) requires a precise process description and terminology as a vantage point for a deeper understanding of its parts.

Thirdly, a precisely defined process model is essential to “transform entrepreneurship from a social science to a science of the artificial” (Venkataraman et al., 2012, p. 21) as proposed by leading entrepreneurship researchers. The level of precision required for artificial science is likely to meet the requirements for advanced research methodology like computer simulation. A well-defined repertoire of processes and parameters is a key requirement for the analysis of complex multi-actor interactions.

The formal model derived to address this research deficit forms the basis of this thesis and is used in all three papers. A first algorithmic representation of Schlüter et al. (2011) focusing primarily on the exchange of pre-commitments served as a venturing point.

**Research deficit 2: The impact of effectual interaction behaviour on the shaping of emerging markets is unclear**

Effectuation has been proposed as a decision logic for the creation of new markets for new ideas (Dew et al., 2010; Sarasvathy & Dew, 2005b; Sarasvathy, 2003). A key instrument for both effectual (Sarasvathy & Dew, 2005b) and non-effectual market creation alike (Hite, 2005; Jack, 2010; Slotte-Kock & Coviello, 2010) is the use of networks. A large body of literature indicates the positive effects of networking (Hoang & Antoncic, 2003) for entrepreneurs and initial empirical findings also support this view for effectuation (Read, Song, et al., 2009). However, the reasons how and why effectuation benefits from the interaction with others in networks remain opaque. The way how new ideas are disseminated in order to shape these emerging markets, as well as the impact that partners and the creator of an idea can have,
are yet to be revealed. Understanding how ideas propagate, why partners are important, and what the limits of influence are, is essential for making effectuation a more reliable toolkit for entrepreneurs. Moreover, these mechanisms need to be understood in order to make effectuation a crucial part of entrepreneurship as a “science of the artificial” (Venkataraman et al., 2012). Effectuation as an individual decision logic is concerned with three entities that impact success and therefore need to be investigated: the process itself, the “network”, and the individual entrepreneur.

Firstly, the impact of the effectual interaction process is not yet sufficiently understood. Dew, Sarasvathy et al. (Dew et al., 2008; Sarasvathy & Dew, 2005b) provided a process-based scheme of effectuation: Dew et al. (2010) enhanced the process by describing transformation methods used by effectual entrepreneurs. Both processes were observed and described on an individual level. However, the impact of individual intersubjective interaction on a macro, i.e. market level is not properly understood yet. Secondly, “the network” has countlessly been recognised as a key success factor in entrepreneurship (Hoang & Antoncic, 2003). Effectual interaction however focuses on intersubjective interaction, i.e. interaction with individual partners rather than “the network”. Therefore, the impact of and the reasons for successful shaping of emerging markets through interaction behaviour of network partners remain largely unknown. Lastly, the individual entrepreneur obviously plays an important role in the effectuation process. However, the extent to which an individual can actually shape an emerging market on its own is unclear.
The first paper therefore engages research deficits associated with the aforementioned entities. Through three research questions it aims to shed light on the impact of process, network, and individual entrepreneur:

1. How does the effectuation process contribute to the shaping of emerging markets?
2. How do effectuators in the network influence the shaping of emerging markets?
3. To which extent can the creator of a new idea influence the shaping of emerging markets in its favour?

**Research deficit 3:** The impact of position in and shape of effectual networks requires further research attention

The positive impact of a favourable network position and shape for entrepreneurs has been confirmed by a large body of literature (see Hoang & Antoncic, 2003). The underlying reasons for successful effectual partnering, however, are yet to be determined (Read, Song, et al., 2009). Moreover, “[a]lmost the entirety of social networks research takes networks as mostly given and outside the control of human action” (Sarasvathy & Venkataraman, 2011, p. 126), an assumption in direct contradiction with the claim of effectuation of an endogenous environment (Sarasvathy & Dew, 2005a). This contradiction warrants a closer look into the mechanisms that let effectuators benefit from networks and actively shape them to fulfil their needs.

With respect to network shape, a widely recognised entrepreneurial networking theory (Burt, 1992) proposes to actively separate partners from each other in order to become a “tertius gaudens” and profit from brokering between the separated parties. This approach, however, significantly differs from the inclusive partnering and co-creation approach of effectuation that is rather following the approach of “tertius iungens” (Obstfeld, 2005) of bringing partners
together for their benefit and participate later. Consequently, a thorough review of the impact of network shape as well as the reasons behind it is required to understand the reasons for success of effectuation despite the contradiction of “conventional wisdom”. This includes the impact of the network shape on the entrepreneur as well as the impact of the entrepreneur on the network shape.

Focusing on the impact of network position and shape as well as the Burt/Obstfeld network development contradiction, the second paper investigates three research questions: (1) How does the network position of an effectuator impact venture performance over time? (2) How does the shape of an effectuator’s network impact venture performance over time? (3) How do effectuators shape their social networks over time?

**Research deficit 4: The interaction of opportunities and effectual entrepreneurs remains opaque**

In their seminal review of entrepreneurship as a science of the artificial, Venkataraman et al. (2012) emphasize the importance of opportunities beyond a mere object developed by entrepreneurs. Instead, they review the findings on the “individual-opportunity nexus” (2012, p. 28). Initially proposed by Venkataraman (Venkataraman, 1997), the individual-opportunity-nexus proposes the joint analysis of individual and opportunity as individuals usually choose opportunities that suit them for various reasons. Neglecting this leads to an incomplete understanding of the entrepreneurship phenomenon (Shane & Venkataraman, 2000). Consequently, Shane and Venkataraman (2000) propose to define entrepreneurship as the “examination of how, by whom, and with what effects opportunities to create future goods and services are discovered, evaluated, and exploited” rather than as “who the entrepreneur is and what he or she does” (both Shane & Venkataraman, 2000, p. 218). Venkataraman et al. therefore
propose further research “looking deeper within simple and direct relationships […] between individual and opportunity (inner and outer environment)” (Venkataraman et al., 2012, p. 28).

The deficit of joint research on opportunity and individual is especially critical for effectuation: the means-principle emphasizes the selection of opportunities which fit “[w]ho I am” and “[w]hat I know” (Sarasvathy, 2001, p. 253). Moreover, effectual opportunity creation is usually influenced by multiple stakeholders in an iterative process. The development of an opportunity is therefore a constant negotiation of personal and other preferences, making it an even more complicated – and less understood – process. A key driver of these complicated negotiations is “environmental complexity” the number of possible combinations of individual preferences in a joint product (Sarasvathy & Dew, 2005a). While effectuation teaches to “embrace contingencies” (Sarasvathy, 2001), i.e. profit from stakeholders with unusual preferences, the impact of environmental complexity and the interplay of individual, opportunity, and environmental complexity are yet to be disentangled.

Focusing on the aspects of environmental complexity as well as the joint development of opportunities over time, the third paper engages the aforementioned deficits by investigating two research questions:

(1) How does environmental complexity influence the creation of opportunities by effectual entrepreneurs?

(2) How do opportunities evolve over time under varying effectual stakeholder behaviour?
1.3 Aspired contribution

This thesis aims to contribute to literature in four ways. Firstly, it presents a formal model of the effectuation process with a detailed review of the interaction of effectuators and stakeholders/market participants. Secondly, it deepens the understanding of effectuation theory by studying the impact of intersubjective behaviour. Thirdly, it introduces network theory, social network analysis, and theory on triads to effectuation. Lastly, it contributes through the utilization of computer simulation. As indicated in Table 1, the formal model and the computer simulation methodology form the basis of all papers. The analysis of behaviour, network, environment, and opportunity creation were executed in different studies.

All research papers contain a detailed description of the literature on effectual processes and the description of the formal model core that is enhanced with additional measurement routines tailored to the respective research endeavour. While the formalization is a contribution in itself, the author is confident that future research will benefit from increased formality as it "enables the elaboration of rough, basic […] theory that is often derived from inductive cases or formal modelling into logically precise and comprehensive theory." (Davis et al., 2007, p. 481). Moreover, the formalised model “provides a different perspective on a research problem, and this fresh look often proves insightful in and of itself” (Harrison et al., 2007, p. 1232).

The first paper aspires to contribute to the understanding of the spread of new ideas in effectual communities. The formal process model is used to investigate the spread of new ideas over time and the effect of transformation methods often applied by effectual entrepreneurs. Moreover, the paper investigates the impact of individual and collective interaction behaviour on the spread of ideas in effectual networks. To contribute a deeper understanding of the reasons that drive the spread of new ideas, the study observes the spread of one particular idea and records the reasons for the dismissal of this idea in each interaction of all participants for varying individual and collective interaction behaviours. The paper contributes a deeper
understanding of effectual partnering processes and provides an updated picture on the importance of partners regarding their behaviour in negotiations and in the propagation of new ideas.

Table 1: Contribution to literature by paper

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Paper I</th>
<th>Paper II</th>
<th>Paper III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development of <strong>formal process and interaction model</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Implementation of formal model as <strong>computer simulation</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Impact of <strong>collective vs. individual interaction behaviour</strong> on spread of ideas in networks</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis of the <strong>reasons for containment of ideas</strong> and the impact on occurrence of interaction behaviour</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact of <strong>network position and shape</strong></td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification of <strong>effectual triad behaviour</strong> (tertius gaudens vs. tertius iungens)</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact of <strong>environmental complexity and interplay of complexity and interaction behaviour</strong></td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development <strong>stages of effectual opportunities</strong></td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The second paper aspires to contribute to the understanding of the effect of networks. The formal model is enhanced with routines to analyse the agents’ network positions and shapes. The paper discusses the impact of network position while treating the network as endogenous, i.e., changeable to the entrepreneur (Sarasvathy & Venkataraman, 2011). In addition the well-known concept of “tertius gaudens” (Burt, 2004) – a strategy to actively keep network contacts apart to benefit from arbitrage and brokering – is discussed and contrasted with the concept of “tertius iungens” (Obstfeld, 2005) – a strategy to actively introduce contacts to each other and
benefit from increased information flow in the network. The study contributes a classification of the effectuation theory according to the gaudens/iungens concepts and provides insights on the benefits of the inclusive effectual partnering approach. In addition, a well-known measure for network shape (constraint) is normalised and applied to identify effectuation-friendly network shapes.

In the third paper, the formal model is applied to study the effectuation process from an opportunity-based perspective. The study contributes an analysis of opportunity creation performance under varying levels of environmental complexity and varying intersubjective behaviours. This contributes to the deeper understanding of effectual interaction behaviour and proposes behavioural requirements especially for complex environments. In addition, the emergence of opportunities is observed from an opportunity-based perspective. This change of perspective allows for strongly requested (Venkataraman et al., 2012) and unique insights on the finalization of opportunities. It proposes a previously unobserved order of phases that an opportunity passes prior to finalization.

In the triangle of contribution to method, theory, and context (2008) this thesis clearly emphasizes the method and theory dimensions. The application of computer simulation is an underrepresented method in entrepreneurship research and the ultimate goal of this thesis is further substantiation of effectuation theory. With respect to the “contribution continuum” ranging from “1) Straight replication” to “8) Develop a new theory that predicts a new phenomenon” this thesis ranges somewhere between “4) Integrative review” and “6) Identification of a new phenomenon”: the review of literature related to the effectuation process and its formalization qualify as an integrative review of effectuation literature. The proposition of a three-staged opportunity finalization or the testing of effectuation theory under boundary conditions leads to propositions not yet covered by effectuation literature and comes close to the identification of new phenomena.
2 Theoretical and conceptual background

The field of effectuation research is relatively young. Firstly mentioned in 2001 (Sarasvathy, 2001), it is still in a nascent/intermediate state (Perry et al., 2012). This thesis aims to deepen effectuation theory focusing on one of the most critical avenues: intersubjective interaction (Sarasvathy & Venkataraman, 2011). This chapter will provide a conceptual background and introduction to literature on the two main theories that were used in this thesis: the theory of effectuation and the analysis of social networks. The review of social network theory is particularly important in the context of effectuation. On the one hand, effectuation relies heavily on interaction. Consequently, a “good network” might turn out beneficial for an entrepreneur using effectual logic. On the other hand, social network researchers presented their idea of how entrepreneurs should use social networks in the past (Burt, 1992; Jack, 2005). These ideas, however, contradict the effectuation process to varying extents, hence, warranting a review of these theories from an effectual perspective.

2.1 Effectuation

Since its introduction in 2001, effectuation research has made significant progress. This chapter will provide a thorough introduction and classification, discuss its key principles and the environment it was designed for. In addition, effectuation is investigated from a process perspective which serves as a foundation for the application of computer simulation.

2.1.1 Introduction and classification

Effectuation is a control-based decision logic designed for the use in highly uncertain environments. The term “control-based” refers to the work of Wiltbank et al. (Wiltbank & Dew, 2006) who classified entrepreneurial approaches with respect to the degree they emphasise “control” and “prediction”. Control in this context means seeing the market as endogenous and
actively shaping future outcomes. Prediction on the other hand refers to the use of planning tools in order to predict future outcomes. As depicted in Figure 2: Control-Prediction matrix, effectuation is classified as strongly emphasizing control. In her seminal paper, Sarasvathy (2001) introduced effectuation as a juxtaposition of “causation”, i.e., prediction-based entrepreneurship. The focus on control and the rejection of prediction delineates effectuation from most other entrepreneurship techniques.

Figure 2: Control-Prediction matrix

Source: (Wiltbank & Dew, 2006, fig. 2)

Effectuation has few theoretical prerequisites: besides the assumption of docility (Sarasvathy & Dew, 2005b), effectuation does not assume trust or certain personality traits (Sarasvathy & Dew, 2008a). Effectuation is teachable: principles and processes can be applied by any entrepreneur; no further skills or genius is required. Effectuation is practical: its
principles and processes were derived from successful serial entrepreneurs, not from a theoretical model.

Especially the last property leads to a noticeable downside of effectuation: effectuation is not complete. It consists of best practices for entrepreneurial action under high uncertainty, but its theory is currently not necessarily exhaustive or self-contained. While research has identified initial relevant dimensions (Chandler et al., 2011), it is not a self-contained theory with a set of axioms and propositions (Sarasvathy, 2001), such as, e.g., transaction cost theory.

2.1.2 Key principles

Both the effectuation process and its principles were presented as a juxtaposition of “classic” prediction-based entrepreneurship approaches, which Sarasvathy referred to as “causation” (2001). The guiding principles of effectuation (see Table 2) exemplify the inherently different approaches of effectuation and causation. They are referred to as “Pilot in a plane”, “Bird-in-hand”, “Affordable loss”, “Crazy quilt”, and “Lemonade principle”.

Pilot in a plane refers to the different approach towards future outcomes. Causation-based entrepreneurship approaches regard the market development as given and proposes to predict the future in order to favourably position their enterprise in this future. Effectuation, however, regards the market development as influenceable and proposes to control future outcomes by controlling the market creation through action – much like a pilot that refuses to trust the autopilot and takes control.

Bird in hand characterises the different approaches regarding available means and goals. While prediction-based entrepreneurship proposes to define a goal first and acquire required resources subsequently, effectuation proposes to use available means including resources and personal preferences as a starting point for venturing. These means should then be used to let
the goal / the opportunity emerge over time. The proverbial bird in hand refers to the available means and the unmentioned pigeon on the roof to a goal.

**Affordable loss** emphasises the alternative investment decision rule of effectuation. While prediction-based entrepreneurs usually choose the opportunity that promises the highest expected returns, effectuation proposes to only pursue opportunities whose loss is affordable in case of failure to ensure long term survival in an uncertain market.

**Crazy quilt** refers to a patchwork quilt that is woven by stitching together pieces of cloth that do not seem to fit, but create a unique and useful outcome. Consequently, effectuation proposes to see other market participants as potential partners that can help to jointly create and define a market. Causation usually sees other market participants as a threat that needs to be analysed, e.g., using the SWOT-approach. This principle also exemplifies the differing attitudes towards networking: effectuation implies a rather dense network that facilitates the exchange of ideas. Causation proposes ego-centric networks to gather resources in order to pursue a predefined goal.

**Lemonade principle**: “If life gives you lemons, make lemonade!” As the English proverb, effectuation proposes to embrace contingencies when they present themselves. Causation, however, calls for action to get “back on track” in the event of unforeseen events in order mitigate these contingencies.

While these five principles are widely accepted in effectuation literature, they are neither universal nor proven to be exhaustive. In scale development, Chandler et al. (2011) propose slightly different dimensions: “experimentation”, “affordable loss”, “flexibility”, and “pre-commitments”. Brettel et al. (Brettel, Mauer, Engelen, & Küpper, 2012) focus on “means”, “affordable loss”, “pre-commitments”, and “contingencies”. Dew et al. (2008) list 12 differences of effectuation and causation. While in line with Sarasvathy’s initial principles, they
extend them with respect to individual action. Perry et al. (2012) propose research along the dimensions mentioned in Table 2. However, they differ from the initially four principles “affordable loss”, “partnering”, “contingencies”, and “control” in Sarasvathy’s initial study (2001). Altogether, these findings underline the nascent state of effectuation theory and highlight the need for consistency and more precise terminology.

Table 2: Principles of effectuation and causation

<table>
<thead>
<tr>
<th>Issue</th>
<th>Causal frame</th>
<th>Effectual frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>View of the future</td>
<td>Predictive. Causal logic frames the future as a continuation of the past. Hence accurate prediction is both necessary and useful.</td>
<td>Creative. Effectual logic frames the future as shaped (at least partially) by willful agents. Prediction is therefore neither easy nor useful.</td>
</tr>
<tr>
<td>Basis for taking action</td>
<td>Goal-oriented. In the causal frame, goals, even when constrained by limited means, determine sub-goals. Goals determine actions, including which individuals bring on board.</td>
<td>Means-oriented. In the effectual frame, goals emerge by imagining courses of action based on given means. Similarly, who comes on board determines what can be and needs to be done. And not vice versa.</td>
</tr>
<tr>
<td>Predisposition toward risk and resources</td>
<td>Expected return. Causal logic frames the new venture creation problem as one of pursuing the (risk-adjusted) maximum opportunity and raising required resources to do so. The focus here is on the upside potential.</td>
<td>Affordable loss. Effectual logic frames the problem as one of pursuing adequately satisfactory opportunities without investing more resources than stakeholders can afford to lose. The focus here is on limiting downside potential.</td>
</tr>
<tr>
<td>Attitude toward outsiders</td>
<td>Competitive analysis. Causal frames promulgate a competitive attitude toward outsiders. Relationships are driven by competitive analyses and the desire to limit dilution of ownership as far as possible.</td>
<td>Partnerships. Effectual frames advocate stitching together partnerships to create new markets. Relationships, particularly equity partnerships, drive the shape and trajectory of the new venture.</td>
</tr>
<tr>
<td>Attitudes toward unexpected contingencies</td>
<td>Avoiding. Accurate predictions, careful planning and unwavering focus on targets for hallmarks of causal frames. Contingencies, therefore, are seen as obstacles to be avoided.</td>
<td>Leveraging. Escalating predictions, imaginative rethinking of possibilities and continual transformations of targets characterize effectual frames. Contingencies, therefore, are seen as opportunities for novelty creation—and hence to be leveraged.</td>
</tr>
</tbody>
</table>

Source: (Dew, Read, Sarasvathy, & Wiltbank, 2009, p. 290 Table 1)

The inherently different approaches and their explicit juxtaposition are not limited to key principles. As depicted in Figure 3 also the application to domains like marketing result in opposing approaches to the same target: “Classic causation” follows the well-known approach to start with “market definition”, “segmentation”, “targeting”, and “positioning” to reach the customer. Effectuation, however, proposes to define multiple markets, add partners and segments, define possible customers, and identify these customers afterwards.
Figure 3: Causation vs. effectuation in market creation

2.1.3 Effectual problem space

Sarasvathy depicts effectuation as an approach to introduce “new products” for “new markets” (Sarasvathy, 2003, fig. 1). Consequently, effectuation is designed to work in environments of high uncertainty. More precisely, the effectual problem space is characterised by Knightian uncertainty, Marchian goal ambiguity, and environmental isotropy (Dew et al., 2008).

Knightian uncertainty (Knight, 1921) describes a level of uncertainty that does neither allow to estimate all possible outcomes of a decision nor to estimate the likelihood of their occurrence. Marchian goal ambiguity implies that “participants in a relationship not only do not know each
other’s motives; they are not quite sure of their own future preferences either” (Sarasvathy & Dew, 2005a, p. 401). Environmental isotropy “refers to the fact that in decisions and actions involving uncertain future consequences it is not always clear ex ante which pieces of information are worth paying attention to and which not” (Sarasvathy & Dew, 2005b, p. 539).

These considerations can be seen as a fact of life and would most likely not be noteworthy to such great detail for empirical work. As this thesis uses computer simulation as research method, these environmental conditions become highly important as they have to be implemented in the formal model of effectuation. Otherwise important prerequisites for the effectual design space would not be met, thus, invalidating the simulation results. Consequently, the formal model used in all three papers pays close attention to the correct implementation of all three preconditions.

2.1.4 Effectuation from a process perspective

Besides guiding principles, understanding effectuation from a process-perspective is imperative for the use of computer simulation. Without a process scheme, there is not much left to actually simulate. Effectuation literature provides three key sources for the creation of an effectual process model: “the two dynamic cycles” (Sarasvathy & Dew, 2005b), the “behavioral theory of the entrepreneurial firm” (Dew et al., 2008), and the description of effectual transformation processes (Dew et al., 2010).

“The two dynamic cycles” (see Figure 4) is the first process description of effectuation. Effectuation is presented as a process that iteratively enhances a project’s resource bases and drives artifact finalization through accumulation of stakeholder constraints at the same time. Moreover, interaction with other individuals and the negotiation of pre-commitments is prominently represented and part of each process cycle.
Figure 4: The two dynamic cycles of effectuation

Source: (Sarasvathy & Dew, 2005b, fig. 1)

The behavioural theory of the entrepreneurial firm as portrayed in Figure 5 enriches the initial process by explicitly associating process steps with theory/principles and the introduction of a key concept: docility. Initially conceptualised by Simon (1990), docility “refers to the tendency to depend on suggestions, recommendations, persuasion, and information obtained through social channels as a major basis for choice” (Simon, 1993, p. 156). Simon added that “[w]e are highly susceptible to social influence and persuasion, susceptibility that I will call docility. I use the term ‘docility’ here in its sense of teachability or educatability – not in its alterative sense of passivity or meekness” (Simon, 1997, p. 41). Unfortunately, despite the theoretical importance no efforts regarding the development of a docility construct or scale for empirical research have been made yet.
Effectual transformations are a rather recent contribution to the process development of effectuation. The overview of Dew et al. (2010) presents mechanisms of opportunity development beyond negotiation with new stakeholders. As presented in Table 3 these transformations are based on best practices of expert serial entrepreneurs and not well integrated into effectuation theory. Those mechanisms are both invoked by external stimuli (e.g., “Deletion and supplementation”, "Free associating") and internal demand for improvement (e.g., “Manipulation”, “Deformation”).
Table 3: Effectual transformation mechanisms

<table>
<thead>
<tr>
<th>Transformation mechanism</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deletion and supplementation</td>
<td>This is the closest to the process of recombination</td>
</tr>
<tr>
<td>Composition and decomposition</td>
<td>The central idea here is reorganizing material that is already there</td>
</tr>
<tr>
<td>Exaptation</td>
<td>The basic idea is that a technique that was originally selected for one trait, owes its later success and survival to another trait which it happens to possess</td>
</tr>
<tr>
<td>Manipulation</td>
<td>Manipulation involves market or product transformations analogous to physically inverting, mirroring, twisting and turning something inside out</td>
</tr>
<tr>
<td>Deformation</td>
<td>Treating the original idea as an elastic platform from which other business ideas are launched</td>
</tr>
<tr>
<td>Localization, regionalization, globalization operations</td>
<td>Transforming the scope of the market by narrowing or enlarging it</td>
</tr>
<tr>
<td>Prototyping</td>
<td>Involves using the original product market as a prototype and then transforming it into a different product that shares the same basic features of the prototype</td>
</tr>
<tr>
<td>Stereotyping</td>
<td>Use of certain simplified or standardized transformation processes that lack originality or inventiveness</td>
</tr>
<tr>
<td>Free associating</td>
<td>Transformations that appear to be essentially idiosyncratic, i.e., based on the experts’ prior knowledge and experience</td>
</tr>
</tbody>
</table>

Source: (Dew et al., 2010, Chapter 5)

2.1.5 Current state of effectuation research

Since its introduction in 2001, effectuation research has made significant progress towards taking the theory from a nascent to an intermediate level (Perry et al., 2012). What began as thought experiments (Sarasvathy, 2001) quickly turned into theory development (Sarasvathy & Dew, 2005b), process definition (Dew et al., 2008) and refinement (Dew et al., 2010). In the meantime adjacent theory was introduced and discussed, e.g., trust (Goel & Karri, 2006), and Austrian economics3 (Chiles, Bluedorn, & Gupta, 2007). The notion of prediction/control-

---

3 Austrian economics: economic phenomena are the result of individual choices
based approaches (Wiltbank & Dew, 2006) allowed a proper classification and showed similarities with and distinctions from other entrepreneurial approaches. In addition effectuation was tested empirically (Read, Song, et al., 2009; Wiltbank, Read, Dew, & Sarasvathy, 2009) and scale development was initiated (Brettel et al., 2012; Chandler et al., 2011). Currently, effectuation is seen as an important building block of turning entrepreneurship into a science of the artificial (Venkataraman et al., 2012).

In the following subchapters we will present the progress of effectuation research in the categories of conceptual and qualitative research and quantitative empirical research.

2.1.5.1 Conceptual effectuation research

Effectuation was first conceptualised by Sarasvathy in 2001. Enhanced versions of principles and process (Dew et al., 2008; Sarasvathy & Dew, 2005b) and additional transformation techniques (Dew et al., 2010) followed over time. Effectuation was conceptualised as prediction-based (Wiltbank & Dew, 2006) and further refined using case studies (Dew, Read, et al., 2009) with debatable results (Baron, 2009). Read et al. (2009) even proposed an application of effectuation to marketing. Given the newness of the concept, Dew et al. prepared an article on “What effectuation is not” (Dew, Read, Sarasvathy, & Wiltbank, n.d. submitted 2011) to discuss and clarify important aspects of effectuation. Literature identified effectuation as a person-centric research approach (Sarasvathy & Venkataraman, 2011; Sarasvathy, 2004a, 2004b) focusing on individual preferences (Dew, Sarasvathy, Read, & Wiltbank, 2009; Read & Sarasvathy, 2005) seeing firms rather as an instrument than a unit of research (Sarasvathy & Venkataraman, 2011).

Besides conceptual development of core theory, the interaction of effectuation and other concepts was discussed intensely. The comparison to Austrian Economics (Chiles et al., 2007, 2008; Sarasvathy & Dew, 2008b, 2009) and trust (Goel & Karri, 2006; Karri & Goel, 2008;
Sarasvathy & Dew, 2008a) led to lively debates regarding conceptual similarities and delineations. Concepts like innovation (Dew & Sarasvathy, 2007), organisational design (Sarasvathy, Dew, Read, & Wiltbank, 2008), March’s technology of foolishness (Sarasvathy & Dew, 2005a), and bricolage (Archer, Baker, & Mauer, 2009; Fisher, 2012), however, were integrated or regarded conceptually close with more ease.

Effectuation strongly benefitted from the discussion in context of other theories. Since 2011, effectuation is also used to contribute to other areas of research. Research regarding disruptive innovation (Chandra & Yang, 2011), innovation in large companies (Svensrud & Åsvoll, 2011), co-creation (Read & Sarasvathy, 2012), marketing (Mort, 2012), and identity (Nielsen & Lassen, 2012) now draw from effectuation rather than informing it.

With respect to conceptual research, scholars are currently discussing how to proceed in the future. The latest research is focusing on the general direction which effectuation research should take (Perry et al., 2012; Sarasvathy & Venkataraman, 2011) and how to integrate it into general entrepreneurship to move forward towards a science of the artificial (Sarasvathy, 2003; Venkataraman et al., 2012) – a designable approach that can be tailored to specific entrepreneurial situations.

2.1.5.2 Quantitative empirical research

While theory development – informed by case studies – started shortly after the introduction of effectuation, the first quantitative study on effectuation was not publicised until 2009. Wiltbank et al. (2009) analysed the impact of control- and prediction-based control strategies in business angel investing using scenario-based questioning. In the same year, Read et al. (2009) presented a meta-analysis of the performance impact of effectual behaviours per principle (“means”, “partnership”, affordable loss”, “contingency”).
The development of an effectuation scale started even later. In 2011 Chandler (Chandler et al., 2011) presented a four-dimensional scale containing one dimension (flexibility) not explicitly mentioned in effectuation theory. A year later, Brettel et al. (Brettel et al., 2012) contributed another four-dimensional scale containing only dimensions explicitly mentioned in theory. The study reveals a positive performance effect of effectuation in highly uncertain environments, i.e., in R&D departments.

Overall, the intensity of effectuation research is impressive. However, leading researchers are now realising that further refinement of the effectuation logic requires a deeper understanding of effectuation theory and effectual behaviour as well as empirical analysis on a level closer to intermediate than nascent state research (Perry et al., 2012). This thesis aims at contributing to the former by integrating available effectuation theory and exploring the mechanisms at work.

2.2 Social networks

The interaction of effectual entrepreneurs through social networks is a key concept of this thesis. Thus, an introduction to social networks and social network analysis is presented here. The concept of a “social network” as a web of ties between individuals was developed by the British anthropologists Barnes (1954), Bott (1955), and Mitchell (1969). Since its introduction, the methods developed by Mitchel using mathematical graph theory have been successfully applied to research areas as diverse as mobility, urbanisation, politics, and belief systems (see Wasserman & Faust, 1994). The basic concepts of social network analysis used in this thesis were drawn from the definitive book on social network analysis by Wasserman and Faust (1994).

In entrepreneurship, the analysis of social network has been used successfully since the 1990s (Greve & Salaff, 2003; Hansen, 1995; Larson & Starr, 1993). A recent review (Hoang &
Antoncic, 2003) reveals significant progress since then dividing the field into the categories “Content”, “Government”, and “Structure”. With respect to structure, Hoang and Antoncic identified two key measures to characterise entrepreneurial social networks: position within the network and shape of the network. While more current literature identified further beneficial uses of entrepreneurial networks (see Jack, 2010) and even developed a rough industry-level development cycle (Slotte-Kock & Coviello, 2010, fig. 1), “theory building is [still] a challenge for entrepreneurship researchers” (Slotte-Kock & Coviello, 2010, p. 34). Effectuation on the other side offers a process model to build and maintain networks from an individual perspective. Therefore, studying the effect of social networks using effectuation processes provides a novel perspective for entrepreneurial social network research altogether.

Within the field of network structure, two prominent theories are often employed as research paradigms: position and shape. Position refers to the position of an individual within a social network. Position is usually measured as degree centrality, i.e. the number of social contacts an individual maintains. Shape is a more advanced construct as it does not only regard direct ties but also the network of ties between known contacts, i.e., indirect ties. Burt (1992) introduced the concept of “structural holes” referring to holes in the social fabric – much like electron holes (Weller, 1967) in solid state physics. Moreover, he introduced “constraint” as a measure to what extent an actor can benefit from a “holey” network shape. Burt’s work also subsumes Granovetter’s often cited idea of the importance of weak, i.e., less used, ties (Granovetter, 1973) under his theoretically and mathematically enhanced “structural holes” paradigm.

The subsequent subchapters will give an introduction to the paradigms of position and shape and highlight the contradiction with current thinking on effectuation.
2.2.1 Analysis of network position

A large variety of beneficial effects has been attributed to a favourable network position in literature. Most findings, however, entertain a common idea: a beneficial network position allows access to resources beyond the entrepreneur’s direct reach. A favourable position increases quantity (Adler & Kwon, 2002; Batjargal, 2003) and variety (Greve & Salaff, 2003; Greve, 1995) of available resources. Consequently successful acquisition of funding (Vanacker, Manigart, Meuleman, & Sels, 2011; Zimmer & Aldrich, 1987), workforce (Freeman, 1999), and information (Semrau & Werner, 2013) is facilitated. Besides tangible resources such as money and goods, intangible resources accessed through networks include information (Birley, 1985), emotional support (Brüderl & Preisendörfer, 1998), production capacity, and distribution channels (Brown & Butler, 1995). While the network is mostly seen as an exogenous resource, some studies also analysed the development of networks as a consequence of entrepreneurial activity (Aldrich & Reese, 1993; Larson & Starr, 1993; Slotte-Kock & Coviello, 2010). However, they focus on market-level development and leave out the actual development mechanisms.

Studies regarding network position usually argue that improved resource access enables entrepreneurs employing prediction-based logics to gather better intelligence on customer preferences and to ease organization of exploration and exploitation of these preferences (Hoang & Antoncic, 2003). Control-based entrepreneurship depends on resources as well. However, it first and foremost employs them to shape an emerging market collaborating with known contacts which provide ideas as well as resources and are part of the emerging market themselves (Dew et al., 2008). Consequently, it is unclear whether more contacts always equal greater success given the higher individual involvement and need for interaction with each contact or stakeholder. Latest empirical effectuation research reveals an inverted-U-shaped
relationship of position (degree centrality) and venture performance (Heuven, Semrau, Kraaijenbrink, & Sigmund, 2011).

Network position is usually measured as centrality. Wasserman and Faust (1994) list degree, betweenness, and closeness centrality as the most common measures. Moreover, they propose the simultaneous use of all three to ensure the validity of findings.

**Degree centrality** denotes the number of known contacts. To compare multiple degree centralities, the measure is usually normalised by the total number of actors within a network.

**Size** was defined by Hoang & Antoncic as “the number of direct links between a focal actor and other actors” (2003, p. 171), which is equal to non-normalised degree centrality.

**Closeness centrality** “focuses on how close an actor is to all the other actors in the set of actors. The idea is that an actor is central if it can quickly interact with all others.” (Wasserman & Faust, 1994, p. 183). Closeness centrality, therefore, measures the average path length to other actors. It is defined as the sum of the length of all geodesics, i.e. shortest paths between focal actor and other actors. The measure is usually calculated as the reciprocal of this sum.

**Betweenness centrality** measures the centrality of an actor with respect to information flow. Assuming that information usually takes the shortest path through a network, betweenness centrality calculates how many shortest paths an actor is part of. Here, a path consists of existing ties between actors. Betweenness centrality notes all shortest paths between all network participants and counts the number of paths the actor under study is part of.

### 2.2.2 Analysis of network shape

Literature reviews on entrepreneurial social networks (Hoang & Antoncic, 2003; Slotte-Kock & Coviello, 2010) identify two key theories with respect to network shape: “weak ties” (Granovetter, 1973) and “structural holes” (Burt, 1992). Granovetter proposed that “bridging
weak ties”, i.e., loose ties with acquaintances otherwise unrelated to other immediate contacts are likely to provide non-redundant information. Granovetter’s ideas were mathematically improved and extended by Burt’s structural holes paradigm. Structural holes describe the absence of network ties between two actors. Building on Simmel’s work on triads (Simmel, 1896) – network configurations of three actors – Burt proposed for entrepreneurs to keep their networks efficient – keep acquaintances separate in order to profit from brokering and arbitrage – and effective – focusing on contacts with many structural holes (Burt, 1992). While Burt’s theory is empirically supported (Baum, Calabrese, & Silverman, 2000; Krackhardt, 1995; Zaheer & McEvily, 1999), the implied behavioural strategy called “tertius gaudens” contradicts the inclusive ideas of effectual social networking warranting further investigation.

Obstfeld (2005) proposed an alternative networking approach called “tertius iungens”. Revisiting Simmel’s work on triads, Obstfeld identified this alternative approach that actively closes structural holes and benefits from collaboration and the emergence of innovation through co-creation. This approach is conceptually much closer to effectuation. While both concepts and resulting network shapes have merits, it is unclear how a network shaped in (dis)favour of a tertius gaudens affects effectual venture performance.

Burt proposes two key measures that will be used in this thesis to measure the shape of networks: efficiency and constraint.

**Efficiency** (Burt, 1992, p. 53) is the degree to which ties of an agent are non-redundant. Thus, in an efficient network, an agent is acquainted with contacts that do not know each other but are well connected themselves. In a network with low efficiency, all known contacts only provide access to agents that can also be reached through other ties as well, rendering the network largely redundant. Burt argues that entrepreneurs benefit from an efficient network.

---

4 Definitions of efficiency and constraint taken from (Jansen, 2013)
Constraint is a measure that captures the extent to which acquainted contacts impede the exploitation of structural holes. Whereas efficiency measures the extent to which direct ties are redundant, constraint measures the degree to which contacts are acquainted. Burt proposes that the acquaintance between known contacts – the absence of a structural hole between them – impedes the application of arbitrage and competition for information. Therefore, a low constraint level indicates an abundance of structural holes resulting in many opportunities for arbitrage and competition for information or resources.

Burt defines $C_i$ as the measure of constraint for actor $i$ as indicated in equation 1.

$$C_i = \sum_{i \neq j} \left( p_{ij} + \sum_{q \neq i \neq j} p_{iq} p_{qj} \right)^2$$

(1)

The term $p_{ij}$ represents the relative share of energy actor $i$ invested in the relationship with $j$. For simplicity, we assume that actors divide their energy equally among known contacts.

Altogether, the literature concerned with social networks of entrepreneurs reveals two key findings that warrant a closer investigation: firstly, it focuses on input-output relationships, thus neglecting the impact of network position and shape on the underlying mechanisms. The disconnect of individual decisions and their consequences for the network requires further investigation, especially for effectuation as a decision logic focusing on individual action. Revealing the individual mechanisms at work and linking them to network-level consequences could enable a more informed, hence, more successful application of it. Secondly, leading social network researchers propose a networking strategy based on keeping network contacts separate (Burt, 1992, 2004). This starkly contradicts the inclusive character of effectuation (Sarasvathy & Dew, 2005a). While there may be more than one strategy to benefit from social networks, a review of these contradicting approaches is necessary to deepen the understanding of the beneficial use of social networks by effectuators.
3 Research methodology

*Simulation is a legitimate, disciplined, and powerful approach to scientific investigation, with the potential to make significant contributions to management theory.*

(Harrison et al., 2007, p. 1243)

This thesis uses agent-based computer simulation as a research method. While computer simulation became a commodity in natural and engineering sciences decades ago, in social sciences this research method is still underrepresented – despite multiple propositions of its adoption in the last decade. Thus, this thesis promotes simplicity and usefulness of computer simulation in this and subsequent research efforts. A detailed introduction of the employed formal model can be found in each paper in part B. Moreover, appendix A gives a quick overview on the structure of the implementation.

3.1 Introduction to computer simulation

Computer simulation is a research method that uses computer-executed formal representations of real-world systems to derive novel insights (Harrison et al., 2007). Most common uses include prediction (weather forecast), optimization (production capacity of factory), and discovery (identification of novel effectuation theory). A formal model is a stylised version of a real-world system that contains algorithmic representations of all key processes and formal data representations of relevant real-world objects (Davis et al., 2007). The focus on key processes and object is necessary to ensure the model is comprehensible and has sufficient explanatory power. Over-simplification, however, leads to “toy models” of either limited explanatory value or limited novel insights (Davis et al., 2007). Three types of computer simulation are most common in social sciences: agent-based models, systems dynamics models and cellular automata models (Harrison et al., 2007).
Agent-based models usually focus on behavioural aspects of multi-actor systems that influence each other through interactions. These models elegantly allow to study the impact of actor-level behaviour on the system. The model consists of one or more actor representations that are instantiated multiple times and simulates their interaction under varying conditions. The clear representation of real-world objects, e.g. customer is represented by customer agent, allow for comprehensive modelling and easy derivation of possible impacts of mechanisms under study in the real world (Harrison et al., 2007).

System dynamics models usually model the system as a whole rather than representing individual actors. They are especially useful when relationships can be modelled as formulas or abstract systematic representations, i.e., arrows and boxes. Consequently, they are the default modelling approach in weather forecast, but can also be used for the study of organizational change or innovations (Harrison et al., 2007).

Besides the theoretical fit, this thesis uses agent-based modelling for another practical reason: with Repast Simphony (North, Howe, Collier, & Vos, 2007) a tool for efficient implementation, verification, and execution of agent-based models is available.

Besides induction and deduction, computer simulation is seen as a “third way of doing science” (Axelrod, 2003, p. 1). It allows the study of complex, intertwined, and non-linear processes beyond empirical feasibility (Harrison et al., 2007) and is especially suited for further development of simple/nascent theory (Davis et al., 2007). Computer simulation allows performing “virtual experiments” (Carley, 1999, p. 2) under boundary conditions and therefore generates insights hardly obtainable otherwise. Consequently, it “capture[s] much more of the complex causality present in typical entrepreneurial settings” (McKelvey, 2004, p. 314). Moreover, computer simulation simplifies data gathering on all levels (individual, system-wide), especially in longitudinal settings. Given that “about half of what people report about
their own interactions is incorrect in one way or another” (Wasserman & Faust, 1994, p. 57) this is an important advantage especially for the study of social networks.

Lastly, computer simulation requires a formal model of the subject under study. These models “serve to keep scientists honest by forcing them to zero in on the most critical variables” (McKelvey, 2004, p. 314) and therefore foster precise definitions and concepts in theory.

Despite its advantages, computer simulation as a research tool has three major drawbacks. Firstly, the model employed represents a stylised version of the real world and is incomplete by necessity. As a consequence, the “correct” modelling is usually at the heart of most critical reviews. Secondly, the output and derived insights are hypothetical and subject to empirical validation before acceptance. Thirdly, the plausible pre-selection of parameter values is critical for valid results. However, empirical results are often hard to translate into abstract parameter values. Consequently, the discussion and careful selection of parameter values is of importance to avoid “garbage in – garbage out” situations.

Given the unique advantages, computer simulations have been successfully applied in entrepreneurship research in the past. In the following, five well published examples that inspired the author during the construction of his model are listed.

Davis, Eisenhardt and Bingham (2009) use stochastic process modelling to investigate the relationship of the amount of structure of an organization, its environment, and performance. They develop a computational model of organization consisting of rules and the environment. Davis et al. reveal that entrepreneurial organizations should build organizational structure in both predictable and unpredictable environments.

Ganco and Agarwal (2009) use an artificial fitness landscape to study the interplay of firm characteristics such as experience, environmental turbulence, and industry live cycle stage. They simulate the co-evolution of all firms under different levels of environmental turbulence
and differences in entry characteristics. Gancho et al. conclude that in highly turbulent environments, diversifying entrants, i.e., firms that existed before market entry, outperform start-ups without previous experience. However, start-ups that learn about these environments later outperform diversifying entrants by far.

Yim (2008) uses a combination of secondary data analysis and computer simulation to investigate rapidly growing start-ups. Using a discrete-choice racing model he demonstrates that focusing on technology and product innovation enabled these start-ups to outperform their competition and that it was not indeed “pure luck”. It is particularly noteworthy that Yim derives part of his simulation parameters from secondary industry data.

Minniti (2005) uses agent-based computer simulation to study the impact of network externalities, i.e., social environment on the level of entrepreneurial activity. Analysing the interplay of wage increase when labour is in demand by entrepreneurs and the ambiguity created by entrepreneurship, she concludes that entrepreneurship agglomerates geographically and that individuals tend to choose what others in their vicinity have chosen.

Minniti (2004) uses agent-based computer simulation to investigate the impact of alertness and available information on the decision to become an entrepreneur. She identifies entrepreneurship as path-dependent and shows that asymmetric information distribution leads to the emergence of entrepreneurship clusters. Using a model borrowed from quantum physics, Minniti reveals that the level of entrepreneurial activity is higher when information is asymmetrically distributed.

3.2 Methodological fit for entrepreneurship research

Literature lists certain prerequisites for theory in order to advance it with computer simulation (Davis et al., 2007; Harrison et al., 2007). As presented in Table 4, five requirements are commonly proposed that need to be fulfilled by the field under study.
The availability of simple theory is an important prerequisite for the application of computer simulation. If no theory is available, formal modelling is impossible or highly ambiguous due to a missing base. If theory is too advanced, simulation is still possible, but unlikely to reveal “novel theoretical insights” (Davis et al., 2007, p. 495). Effectuation is in such a nascent state (Perry et al., 2012). Moreover, a basic understanding of micro-level processes is required. Otherwise, there is too much ambiguity regarding the agents’ tasks in a simulation. For effectuation, sufficient actor-level process descriptions exist (Dew et al., 2008, 2010; Sarasvathy & Dew, 2005b) though this area is still subject to research (Sarasvathy & Venkataraman, 2011).

As for all research endeavours, an intriguing research question is of the essence. For the application of computer simulation, however, Davis et al. recommend questions targeting “substantial theoretical issue” (Davis et al., 2007, p. 483). Effectuation fulfils this criterion as demonstrated in chapter 1.2. In addition, the data required to answer these questions should be hard to obtain, otherwise the use of empirical research is preferable. This is true for effectuation as well, given for example the lack of scales for docility, or the need for detailed data on social network to compute the measures proposed by Burt. Moreover, the process is highly non-linear and based on intertwined multi-agent interaction, which is hard to capture by empirical methods.

Lastly, computer simulation is required to end at a certain point in time. While this end can be predetermined, e.g., “three days” for weather forecasts, an equilibrium state is preferable. Fortunately, the effectuation process provides an end for opportunity development as effectuators can “[d]eclare the effectual transformation complete” (Sarasvathy & Dew, 2005b, p. 549).
Table 4: Requirements for the application of computer simulation

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Fulfilment by effectuation theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of “simple theory”</td>
<td>According to Perry et al. (2012) effectuation is in a nascent/intermediate state</td>
</tr>
<tr>
<td>(Davis et al., 2007, p. 482)</td>
<td></td>
</tr>
<tr>
<td>Understanding of “micro-level processes”</td>
<td>Sufficient process descriptions given (Dew et al., 2008, 2010; Sarasvathy &amp; Dew, 2005b)</td>
</tr>
<tr>
<td>(Harrison et al., 2007, p. 1231)</td>
<td></td>
</tr>
<tr>
<td>Intriguing research question regarding substantial theoretical issue</td>
<td>Impact of intersubjective interaction behaviour fundamental yet unclear (Perry et al., 2012; Sarasvathy &amp; Venkataraman, 2011; Venkataraman et al., 2012)</td>
</tr>
<tr>
<td>(Davis et al., 2007, p. 483)</td>
<td></td>
</tr>
<tr>
<td>Difficulty of empirical measurement</td>
<td>Lack of docility scale despite theoretical importance (Dew et al., 2008) and challenging acquisition of network data (Wasserman &amp; Faust, 1994). Moreover, multi-agent, non-linear, intertwined processes at work</td>
</tr>
<tr>
<td>(Harrison et al., 2007, p. 1230)</td>
<td></td>
</tr>
<tr>
<td>Fulfil “equilibrium assumption”</td>
<td>Effectuators can “[d]eclare the effectual transformation complete” (Sarasvathy &amp; Dew, 2005b, p. 549) providing a stable state</td>
</tr>
<tr>
<td>(McKelvey, 2004, p. 317), i.e. simulation can reach a stable state</td>
<td></td>
</tr>
</tbody>
</table>

Source: Own illustration

3.3 Simulation creation process and nature of outcomes

Davis et al. (Davis et al., 2007) offer a clear roadmap for the creation of a simulation for theory development. Using the proposed process provides interesting, relevant, and valid research using computer simulation. Starting with an intriguing research question, researchers should subsequently identify simple theory and choose a suitable simulation approach. Afterwards, a formal model needs to be created and validated. Once the model is complete, experimentation (including validation) can start.
The most crucial part of computer simulation research is the development of the formal representation of the real world. It is more of an art than a skill and needs to balance out the demands for simplicity and comprehensiveness, for understandability and complexity (Harrison et al., 2007). It has to be simple, so proposed effects can be unambiguously attributed to certain inputs and processes. It has to be comprehensive in a way that all vital parts of theory are represented. It has to be understandable to be regarded as valid by the researcher community. Lastly, it has to be complex enough to produce counter-intuitive and novel insights into the underlying theory. Consequently, formal modelling itself is already a contribution as it “constitutes an exercise in theory development” (Harrison et al., 2007, p. 1233) and “serve[s] to keep scientists honest” (McKelvey, 2004, p. 314).

In a broader context, simulation alone is insufficient to further theory development. As visualised in Figure 6, computer simulation experiments are part of an iterative cycle and symbiotically coupled with empirical research. Computer simulation is informed by theoretical and empirical research and delivers propositions that, in turn, inform theory development and empirical research to validate the propositions of computer simulations.

**Figure 6: Interaction of computer simulation and theory development**

Source: (Harrison et al., 2007, fig. 1)
4 Summary of Research Papers

This thesis consists of three research papers on the effects and implications of effectual intersubjective interaction. This section provides a summary of each paper and points out important contributions individually. The complete papers are available in Part B.

All papers are built on a jointly used computer simulation of effectuation. While the core stayed the same throughout the research effort, the model was extended by more sophisticated measurement or initialization functions for each paper. Given the controversial reception of studies that “only” provide a formal model (and consequently lack data analysis of any kind) at research conferences, the author refrained from drafting a methodological study first and subsequently referencing it. Instead, the model is presented and motivated in each paper.

Paper I focuses on the impact of individual and collective effectual interaction behaviour on the creation of new markets for an idea. It reveals the importance of docile and persistent partners and the surprisingly small impact an individual entrepreneur can have. Moreover, it underlines the importance of transformation mechanisms for effectual logic.

Paper II is concerned with the impact of network position and shape on effectual venture performance. Using centrality measures drawn from social network analysis and Burt’s measures of network shape (Burt, 1992), it reveals the limited importance of centrality at the beginning of venturing. Moreover, it shows that Burt’s idea of efficient and effective network correlate negatively with effectual performance.

Paper III provides an opportunity-based perspective on effectual entrepreneurship. Studying the interplay environmental uncertainty and intersubjective interaction, it reveals the increasing importance of docility for the successful development of more complex opportunities in more complex environments. In addition, it reveals an order in which effectual
entrepreneurs complete an opportunity: firstly, the core features are fixed, secondly, additional features are added, and thirdly, the stakeholder group is finalised.

4.1 Paper I: Individual vs. Collective Control in Effectual Social Networking: A Simulation Study

Paper I investigates on the impact of process, individual, and collective behaviour of entrepreneurs using effectual logic on new market creation.

What do we know? Social networks are of high importance for successful entrepreneurs (Hoang & Antoncic, 2003). A large body of network-related studies reveals that networks serve as access channels to resources and thus improve venture performance. Effectuation as an alternative decision logic also heavily relies on interaction with other market participants. However, it proposes a different approach to networks focusing on co-creation and control of market development (Sarasvathy, 2001).

What do we not know? As the basic principles of effectuation are widely accepted, the mechanics for their market-level impact become the focus of research attention, especially the impact of intersubjective interaction (Sarasvathy & Venkataraman, 2011). It is unclear, how actor-level interaction impacts market-level developments. Docility was identified as one important component of interaction behaviour (Dew et al., 2008). However, the reasons for market-level impact are yet to be determined. Moreover, it is unclear which other behavioural parameters are relevant. In addition, transformation processes were only recently recognised as an important building block of effectuation (Dew et al., 2010). Consequently, the mechanics of market-level impact are opaque.
**Why is it important to know it?** Understanding the inner works of effectuation and the impact of actor-level behaviour can have on markets is critical for multiple reasons. Firstly, effectuation emphasises the joint development of opportunities while prediction-based entrepreneurship usually portrays “the entrepreneur” as the one mastermind that coordinates all activity. A review of these contradicting views will put the impact of “the entrepreneur”, stakeholders and the process into perspective. Secondly, to support the effort of developing entrepreneurship as a science of the artificial requires a detailed understanding of processes and consequences. In order to become an important building block, effectuation has to be studied in the aforementioned detail.

**What are we doing about it?** Using computer simulation, this paper analyses the complex, non-linear multi-actor processes at work and determines the impact of process, i.e., transformation, individual, and collective interaction behaviour. Using the creation of new markets as scenario, the simulation tracks the dissemination of a specific idea in the network of effectuators and end customers. It tests the impact of parameter variations, (dis)allowing transformation processes while tracking the dissemination of the idea under study as well as reasons for non-dissemination.

**What do we contribute?** Paper I makes four contributions to literature. Firstly, it contributes a formal model of effectuation. The model unifies available descriptions of the effectuation process, effectual principles, and views. It allows the study of intersubjective interaction and its consequences on arbitrary levels of detail even under boundary conditions. Besides docility, it reveals the persistence to approach potential stakeholders as another relevant behavioural parameter. Secondly, this paper reveals the importance of transformation mechanisms for effectuation. Allowing the propagation of new ideas beyond project borders has a “trigger effect” on the creation of new markets: Although rarely used, it significantly increases the adoption rate of the idea under study. Thirdly, this paper contributes the
significance of collective interaction behaviour. Both the levels of collective docility and persistence have a significantly positive impact on the creation of a new market for the idea under study. Moreover, tracking the reasons for non-dissemination of the idea under study reveals that high docility improves negotiation success and helps to include customer preferences more successfully. Lastly, Paper I contributes the relatively little impact of the interaction behaviour of an individual. Increased persistence to approach stakeholders has little, increased docility no measureable impact on the creation of new markets. “This contrasts with the image of the persistent entrepreneur who holds on against all odds and against all skepticism to bring an idea to fruition” (Wood & McKinley, 2010, p. 71).

4.2 Paper II: Entrepreneurial Mingling Secrets: Investigating the Performance Impact of Network Structure for Control-based Entrepreneurship using Agent-based Simulation

Paper II investigates the impact of network position and shape on effectual venture performance.

What do we know? Networks are an essential tool for successful entrepreneurs (Slotte-Kock & Coviello, 2010). A large body of literature lists countless advantages of being “well connected”. Most of these studies entertain a common theme: networks extend the reach of entrepreneurs thus making resources accessible otherwise beyond their grasp (Hoang & Antoncic, 2003). Picturing “the entrepreneur” as the centre of action, these studies usually argue and show that being in a more central position increases quantity and variety of resources an entrepreneur can access (Jack, 2010). With respect to shape, network literature proposes for entrepreneurs to apply the “tertius gaudens” approach, keep acquaintances separate, and benefit from brokerage and arbitrage (Burt, 1992).
What do we not know? Effectuation proposes a networking approach radically different from “classic” prediction-based entrepreneurship (Sarasvathy, 2001). Consequently, it is unclear whether network position plays an equally significant role for effectual entrepreneurs and how the effectual process uses it. Moreover, the “tertius gaudens” approach proposed for entrepreneurs by Burt (1992) directly contradicts the inclusive partnering approach of effectual entrepreneurship. Consequently, an analysis of shape and performance as well as network development over time is required. In addition, an alternative theoretical foundation is required for effectual networking.

Why is it important to know it? Given the prediction-based conditioning of entrepreneurial network research, the development of a control-based alternative broadens the view on entrepreneurial networking and the range of options for practical application. Moreover, the lack of process-oriented studies on the use of social networks effectively blocks effectuation research – and entrepreneurship research in general – from developing entrepreneurial and effectual network theory. In addition, Burt’s tertius gaudens approach proposes a networking methodology that contradicts the effectual approach more or less directly. Consequently, an analysis of these contradictions is required to understand how and why effectual networking works. Moreover, this analysis is required to unravel the contradicting empirical evidence of successful effectual networking on one side and Burt’s “tertius gaudens” approach on the other side.

What are we doing about it? Based on the formal model of effectuation, paper II uses computer simulation to measure the network position and shape of each agent in each time step of the simulation. In addition, the simulation measures the level of demand satisfaction, the extent to which an agent’s business opportunity is compatible with all other agent’s conscious demands (reflected in their respective business opportunities). Moreover, a literature search reveals an opposing theoretical networking foundation called “tertius iungens” (Obstfeld,
This approach proposes the creation of a dense network in order to profit from the free flow exchange of ideas and information.

**What do we contribute?** Paper II contributes three major insights regarding the impact of network position and shape of effectual entrepreneurs.

Firstly, demand satisfaction is positively related with degree centrality, i.e. the number of known contacts. While the initial degree centrality has a limited positive impact, the final degree centrality is clearly correlated with demand satisfaction. Consequently, effectuators profit from good initial positions, however, they can mitigate the disadvantage of initial position to a great extent. However, being well connected afterwards as a result of good networking has a clearly beneficial impact on success. Consequently, initial betweenness centrality, i.e. being on the shortest path between many individuals, means a beneficial starting point for effectuators. However, if an effectuator’s betweenness centrality is still high in the end, she or he was not able to become obsolete by weaving a dense network around her or him. Consequently, final betweenness centrality is negatively associated with performance.

Secondly, paper II shows that low efficiency and high (normalised) constraint are positively related with performance for effectual entrepreneurs. Consequently, Burt’s proposed negative impact of these measurements does not hold and rather confirms Obstfeld’s “tertius iungens” approach.

Thirdly, this paper contributes a longitudinal analysis of degree centrality, betweenness centrality, efficiency, and (normalised) constraint. Based on the aforementioned arguments, this analysis confirms the increase of degree centrality and decrease of betweenness centrality due to the effectual networking process. Moreover, the analysis reveals that the effectual networking approach decreases network efficiency and increases normalised constraint over time. These results indicate that entrepreneurs using effectual logic not only profit from a network position
and shape suited for “tertius iungens” approach, but actively shape their network in a way that these shapes emerge.

4.3 Paper III: Out of Thin Air? Simulating Entrepreneurial Opportunity Creation and the Impact of Environmental Complexity and Stakeholder Behavior

Paper III investigates the interplay of environmental complexity and stakeholder behaviour on opportunity creation, with respect to success and the emergence of patterns in the creation process.

What do we know? Effectuation is an alternative decision logic for the creation of opportunities under high uncertainty. Derived from experiments with expert serial entrepreneurs it provides an entrepreneur-focused process for the interaction with other market participants and the joint creation and manipulation of opportunity and market at the same time (Dew et al., 2008).

What do we not know? Current entrepreneurship research usually focuses on entrepreneurs and pays less attention to the opportunity itself (Shane, 2004). Consequently, the opportunity as well as the interplay of opportunity and entrepreneur has received insufficient research attention (Venkataraman et al., 2012). Research regarding opportunities still targets basic questions. Currently, a lively debate whether opportunities are created or discovered is still questioning most essential assumptions (Alvarez & Barney, 2013; Shane, 2012). The interplay of opportunity and entrepreneur is similarly under-researched. While the impact of complexity of environment/opportunity (Sarasvathy, 2004a) or the impact of entrepreneurial behaviour (Dew et al., 2008) have been discussed separately, the interplay of both have neither been researched nor been theorised about yet.
Why is it important to know it? Both the emergence of opportunities and the interplay of opportunity are mentioned as critical areas of research for the development of entrepreneurship into a “science of the artificial”, a toolbox for entrepreneurs that can be tailored to specific entrepreneurial scenarios (Venkataraman et al., 2012). While the creation or discovery of opportunities are the causes of entrepreneurship in the first place, few studies are concerned with the development of opportunities from an opportunity perspective. This limited perspective obscures the understanding of opportunity development and thus hinders effectuation research. Moreover, entrepreneurship needs to be understood as the interplay of entrepreneur and opportunity (Shane, 2004). Otherwise, the underlying interplay, hence the mechanics at work, remain opaque, blocking the way to a deeper understanding of when to best use which entrepreneurial strategies, and why.

What are we doing about it? Using computer simulation, we analyse the interplay of entrepreneur and opportunity as well as the emergence of opportunities in two series of experiments. Starting with the interplay of entrepreneur and opportunity, we analyse the interplay of entrepreneurial interaction behaviour and the complexity of opportunity/environment. Using demand satisfaction as performance measure, we test the impact of various levels of docility and various numbers of available features and variants at the same time. Our second series of experiments measures the time to completion of three stages of opportunity development.

What do we contribute? Paper III contributes two discoveries regarding opportunity development using effectuation: firstly, it reveals different impact of different kinds of complexity. While increased numbers of features have no impact on successful opportunity creation, the increased availability of alternative variants per feature does significantly decrease successful development of opportunities. Moreover, the negative impact is strongly dependent on effectual interaction behaviour. Setups with highly docile effectuators experience
significantly less negative impact of increased availability of alternative variants per feature than setups with non-docile effectuators. Secondly, the evolution of opportunities follows – regardless of effectual behaviour – a fixed scheme: core completion, feature completion, stakeholder completion. Core completion refers to the fixation of variants for available features. While initial interactions create change in variants of features, these changes are completed first resulting in a fixed core. Next, the addition of new features is finalised. These new features are the result of further interaction. Given the balance of power in negotiations, new stakeholders can add new features easily, but hardly change the opinion of a whole group of incumbent stakeholders regarding previously negotiated features. Lastly, the stakeholder group is completed. While in theory these last stakeholders could still change the opportunity completely, they find themselves mostly in a “take it or leave it” position. Interestingly, this order opportunity completion does not change regardless of, e.g., the general level of docility.
5 Overall implications

The research presented in this thesis contributes to literature in many theoretical and – to a lesser extent – in practical ways. Moreover, it opens up important areas of further research for quantitative empirical confirmation, qualitative empirical understanding of process details, and formal/mathematical model improvement.

5.1 Theoretical implications

Beyond the conclusions and implications presented in chapter 1, the research presented in this thesis has broader implications for the fields of effectuation research, entrepreneurship research, and research in adjacent theories. Moreover, this thesis identified areas of further research requiring qualitative, quantitative, and computer simulation methodologies.

The overall implications of this thesis for effectuation research are manifold. The development of a formal model, the impact of individual and collective behaviour, the analysis of transformation, the impact of network position and shape, the tertius iungens approach, the interplay of environment and interaction behaviour, and the analysis of opportunity creation have broad implications beyond their individual contributions.

The developed formal model of effectuation serves as starting point for the improvement of terminology and process descriptions. Formal models “serve to keep scientists honest by forcing them to zero in on the most critical variables” (McKelvey, 2004, p. 314). Consequently, formal modelling helps to overcome the current nascent state of effectuation (Perry et al., 2012). Focusing on intersubjective interaction, the formal model employed in papers I – III already contributed to the deeper understanding of docility and identified “persistence” as an important part of entrepreneurial intersubjective interaction. Moreover, the formal model of effectuation enables the analysis of market-level consequences of actor-level behaviour and provides
detailed theory-based reasoning for them. While most studies either study individual action (Dew et al., 2008, 2010; Dew, Read, et al., 2009; Sarasvathy & Dew, 2005b; Sarasvathy, 2001) or market impact of effectuation (Brettel et al., 2012; Chandler et al., 2011; Read, Song, et al., 2009; Wiltbank et al., 2009), the formal modelling approach provides a unique perspective to relate both levels beyond empirical feasibility and foster theory development. In addition, the provided formal model allows the study of effectuation under boundary conditions hardly found in reality. These “virtual experiments” (Carley, 1999, Chapter 2) allow data collection in extreme cases, which is nearly impossible in empirical research. While these cases have limited importance for real world scenarios, they are essential for theory development (see Davis et al., 2009). Altogether, the formal model developed in this thesis provides a novel and fruitful approach for the development of effectuation theory.

The analysis of individual and collective intersubjective interaction behaviour yields implications for effectuation research. First of all, the analysis underlines the view of “docility as a fundamental behavioral construct” (Dew et al., 2008, p. 49). In addition, it deepens the understanding of its application in effectuation and differentiates between individual and collective impact of this actor-level parameter on market-level impact. Given the limited individual and significant collective impact of intersubjective interaction parameters, paper I implies a paradigm shift from ego-centric to group-based research on effectuation. It contracts the all-too heroic view of “the entrepreneur” as the single source of success, single-handedly orchestrating passive stakeholders and stoically pushing through ingenious ideas. In line with literature (Wood & McKinley, 2010) these results rather propose to see effectuation as a collaborative process where collective benevolence trumps individual stubbornness.

The analysis of transformation holds several implications for effectuation as well. This thesis contributes a first rough formal process and provides a link to intersubjective interaction. Thereby it sets a starting point for further differentiation of transformation processes and their
inclusion into the effectuation process. The overwhelmingly positive impact of transformation demonstrated in paper I underlines its “practical” importance for effectuation and calls for a reinterpretation: beyond the optimization of an opportunity (Dew et al., 2010), the inclusion of external ideas via transformation is a mechanism to disseminate ideas in order to create a market for them.

The longitudinal analysis of network position and shape holds crucial implications for the further development of effectuation theory. It analyses the complex bidirectional interplay of position, shape and intersubjective interaction. Thus, both the development of networks through effectual processes and the impact of networks on effectual processes is demonstrated and explained. Contributing integrated “process- and outcome-oriented research” (Hoang & Antoncic, 2003, p. 165) this thesis fosters theory development sorely lacked on network (Hoang & Antoncic, 2003; Jack, 2010; Slotte-Kock & Coviello, 2010) and individual (Sarasvathy & Venkataraman, 2011) level. In addition, the inclusion of the “tertius iungens” networking approach (Obstfeld, 2005) broadens the effectual theory foundation and poses yet another difference between effectuation and causation. Moreover, it delivers a successful application example of Obstfeld’s theory and shows that Burt’s measures of efficiency and (normalised) constraint (Burt, 1992) work to identify networks suitable for effectuators. It also reasons why the tertius iungens approach works and explained mixed empirical results finding successful applications for both models (see Burt, 2004; Obstfeld, 2005).

Lastly, the interplay of environment and interaction behaviour and the order of opportunity development – analysed in paper III – contributes a novel perspective to effectuation research. Available effectuation literature is usually actor-focused treating opportunities merely as the output of the actor-based entrepreneurship process. This approach, however, neglects the nexus of individual and opportunity (Sarason, Dean, & Dillard, 2006; Shane & Venkataraman, 2000; Venkataraman et al., 2012). The “nexus of individual and
opportunity”-idea proposes that individual and opportunity require a certain level of congruence – otherwise they do not fit. In 2012, Venkataraman et al. proposed to develop this nexus further and research in the “[a]ction and [i]nteraction” (Venkataraman et al., 2012, p. 28) of entrepreneur and opportunity. This study provides a significant contribution in this direction by analysing the interaction of entrepreneur and opportunity for various levels of environmental complexity and interaction behaviour (docility).

Consequently, this research also contributes to the advancement of entrepreneurship research in general. Effectuation was proposed as one important entrepreneurial technique (Venkataraman et al., 2012) and the deeper understanding as well as the formalization, thus contributing to the development of entrepreneurship into a science of the artificial. The development of an artificial science requires both a deep understanding and a high level of formalization of the employed techniques. These preconditions are necessary to devise optimal processes and parameter choices, e.g. required level of docility, for specific entrepreneurial scenarios. In addition, this study contributes to the development of the “entrepreneurial method” (Sarasvathy & Venkataraman, 2011, p. 113) by contributing a deeper understanding of the intersubjective. While most entrepreneurship studies focus on individuals or teams, “no journal article on details of the numerous relationships and deals that entrepreneurs routinely negotiate with a wide variety of stakeholders” (Sarasvathy & Venkataraman, 2011, p. 126) is available. Hence, knowledge on this aspect fosters understanding and theory building for entrepreneurship.

Moreover, a deeper understanding of effectuation also fuels the discussion regarding the validity and normative superiority of “classic” prediction-based and control-based entrepreneurship (Baron, 2009; Chandler et al., 2011; Sarasvathy, 2001; Wiltbank & Dew, 2006). While the positive impact of effectuation has been shown empirically for situations of high uncertainty (Brettel et al., 2012; Chandler et al., 2011), it is still unclear what other factors
promote or demote the application of either prediction- or control-based entrepreneurship. Assuming that Burt’s examples of “tertius gaudens” entrepreneurship (Burt, 1992, 2004) focused on prediction-based entrepreneurship, this thesis contributes network position and shape as an environmental condition that impacts normative superiority.

This thesis also has implications for adjacent fields of research, namely social network theory, social network analysis, and organizational innovation. Social network theory often takes “networks as mostly given and outside the control of human action” (Sarasvathy & Venkataraman, 2011, p. 126). Hoang and Antoncic propose further research on “how network […] structure emerge[s] over time” (Hoang & Antoncic, 2003, p. 165) and criticise a lack of process-oriented studies. The longitudinal analysis of the impact of position and shape based on a formal model of the effectuation process contributes novel insights that can help form a theory on the development, growth, and impact of social networks especially for entrepreneurship.

Paper II uses Burt’s efficiency and constraint measures to analyse the impact of network shape on effectual success. Burt’s constraint measure precisely reflects two ways to reduce constraint: keep acquaintances separate and increase the number of known contacts. However, it is hardly suited to compare the degree of constraint-ness of two ego-networks of different sizes. With exactly that task at hand, paper II proposes a method for the normalization of Burt’s constraint measure: the normalised constraint. It normalises the number of known contacts and makes the relative constraint-ness of two actor-networks comparable.

Lastly, this thesis contributes to the originating field of Obstfeld’s “tertius iungens” approach (Obstfeld, 2005), organizational innovation. While the application of the “tertius iungens” networking strategy qualifies effectuation for the development of innovative opportunities, effectuation also reflects back on organizational innovation: it strengthens
Obstfeld’s arguments by providing numerous successful practical examples of the tertius iungens methodology (e.g. Read, Song, et al., 2009) and a complete theoretical concept (Sarasvathy, 2001) that is based on it. Moreover, the successful application of effectuation in R&D-departments (Brettel et al., 2012) and its connection to the development of innovation through tertius iungens may imply that the development of highly innovative products is – besides high uncertainty – another antecedent for normative superiority of effectuation.

5.2 Areas of further research

The findings presented in this thesis warrant further research regarding effectuation. Besides a validation of the theoretical propositions, paper I-III open individual new areas of research both theoretical and empirical. Moreover, the formal model itself holds important areas of further research in itself, both theoretical and empirical as well.

Firstly, an empirical validation of the propositions presented in paper I-III is in order. While simulation can contribute to theory development and postulate relationships, an alignment with the “real world” is required to see whether these propositions hold. Given the comprehensive discussion of each proposition, computer simulation can simplify empirical research as it provides strong hypotheses. Considering the often non-linear nature of relationships, this is a strenuous undertaking; however, it is necessary to validate the proposed findings and the formal model of effectuation.

Prior to empirical validation, scale development for docility and persistence is required. The persistence parameter is conceptually close to the “proactiveness” construct of entrepreneurial orientation (Lumpkin & Dess, 1996), thus the items proposed for it might be a first proxy. Despite its conceptual importance, so far the development of a scale for docility has
not taken place at all\(^5\). Given the significant recent development of effectuation scales (Brettel et al., 2012; Chandler et al., 2011) and the significant theoretical work on docility (Dew et al., 2008; Dew, 2003; Simmel, 1896), the need for such a scale goes beyond this thesis.

Besides empirical validation, papers I-III opened up individual research areas – theoretical and empirical – with respect to individual and collective behaviour (paper I), transformation mechanisms (paper I), network position and shape (paper II), and environmental complexity and opportunities (paper III).

Paper I analysed the impact of collective and individual intersubjective behaviour, focusing on either one or all but one actor. Consequently, the study of the impact of groups with respect to their size and their group behaviour on the creation of new markets could provide further details in this area of research. Is there a minimum group size required in order to have significant impact? How does the level of docility and persistence of groups impact new market creation and how does group size moderate it? In addition, further qualitative empirical research is necessary to better understand the negotiation process itself and the behavioural parameters relevant for it. As “[e]ven the literature that is directly focused on negotiations has mostly neglected new venture creation processes” (Sarasvathy & Venkataraman, 2011, p. 126), a deeper understanding is essential to further theory in this crucial area.

Paper I also investigated the impact of effectual transformation processes. The results underline the importance of these techniques. While the formal model only incorporated one abstract method, the detailed study and incorporation of further methods could contribute to the works of Dew et al. (2010), helping to evaluate when to use which method most effectively. In addition, qualitative research is required to understand process parameters for each method.

\(^5\) Excluding the works on docility of cattle (Burrow, Seifert, & Corbet, 1988)
Paper II investigated the impact of network shape and size of individuals opening multiple areas of further research. Firstly, the paper focuses on individuals, leaving the analysis of group impact to further research. While social analysis provides measures for the position of groups as well (Everett & Borgatti, 1999), the concepts of shape (efficiency and constraint) are currently not applicable to groups necessitating a network theoretical endeavour beforehand. Secondly, the formal model assumes equal strength of all ties between all actors. In reality, however, effectuators are likely not to interact to an equal degree with all their contacts. Consequently, an investigation on the effects of tie strengths could contribute to the development of entrepreneurial networking theory, also adding a new perspective on the importance of weak ties as proposed by Granovetter (1973). While measures of position and shape already support different tie strengths, empirical research regarding tie strength and impact on behaviour is required to improve the relevance of this line of research. Lastly, the theoretical stereotypes “tertius gaudens” (Burt, 1992) and “tertius iungens” (Obstfeld, 2005) warrant further research. While leading effectuation researchers generally question the compatibility of Burt’s ideas and effectuation (Sarasvathy & Dew, 2005a, 2005b), both Obstfeld and Burt (Burt, 2000) argue that superiority of these strategies is conditional. Consequently, the identification and description of a possible application scenario of “tertius gaudens” for effectuation is an area for both empirical and theoretical research.

Paper III studied the interplay of complexity and effectual intersubjective interaction behaviour in order to analyse their impact on opportunity creation. Moreover, the creation phases of opportunities were analysed longitudinally. Both experiments opened up further areas of research. Firstly, the interplay of collective docility and environmental complexity repositioned the idea of docility from a merely personal attitude towards a free parameter for optimizing the effectual approach. While this is definitely in line with the idea of “entrepreneurship as a science of the artificial” (Sarasvathy, 2003; Shane, 2012; Simon, 1996)
the almost threshold-like decrease of success in opportunity development due to docility calls for further investigation. Firstly, through case studies regarding repeated interaction in more or less docile environments, and subsequently theoretical with more detailed computer simulation. The combined approach is necessary to detect the deeper reasons for this step-change in performance. Secondly, the emergence of order in opportunity creation is a novel (theoretical) observation. Consequently, it opens up new areas of research in opportunity creation: How does interaction behaviour impact (not change) the speed of order creation? How do different negotiation processes impact creation order? Does this or another stable creation order also emerge from other entrepreneurial techniques? Again, both theoretical and empirical research is needed for theorizing, validation, and the gathering of facts for model building.

Besides the areas of research informed by the results of paper I-III, an equally important area of research was informed by the formal model itself: the affordable loss conundrum. The implementation of the affordable loss criterion poses three challenges, yet to overcome by effectuation research: process implications, optimization implications, and resource implications. Firstly, while the idea of keeping the loss affordable seems straightforward, the process of doing so is quite unclear. While extant literature analysed the implications of affordable loss to great extent (Dew, Sarasvathy, et al., 2009), the processes of determination, adaptation as well as relevant parameters for affordable loss levels remain opaque. The level of affordable loss is simply attributed to an individual decision. Consequently, qualitative empirical research regarding process and parameters is a prospective area of further research. Secondly, the affordable loss principle was formulated as a juxtaposition of the prediction-based idea of returns maximization. However, the affordable loss principle does not contain any superlative, hence is a constraint, not an optimization criterion. Hence, the ultimate goal of effectuation is unclear. Additional theoretical research is required to present an alternative optimization criterion. Paper I speculates that the maximization of market size may be such a
criterion, but the presented arguments can only serve as a starting point. Lastly, from a formal model point of view the implementation of the affordable loss criterion requires the implementation of a complete resource life-cycle simulation. This includes the attribution of costs to actions. The multitude of actions require a precise selection of “correct” values and ratios to ensure that the simulation does not produce invalid results. Consequently, a significant research effort regarding resource consumption of effectual action is required beforehand. Moreover, simple questions such as “Under which conditions is effectuation/causation more resource-efficient?” require the calibration of costs among various entrepreneurship approaches as well.

5.3 Managerial implications

As a theoretical study, the number of practical implications is limited. Moreover, the presented results of virtual experiments are yet subject to empirical validation. However, three practical implications for entrepreneurs considering the use of effectual logic are to be considered:

Paper I presented the difference of collective vs. individual impact on the creation of new markets. From a practitioner’s perspective, this implies that the search for a proper environment is paramount. Regardless of the amount of potential stakeholders, practitioners should ensure an environment that is sufficiently docile, otherwise the creation of a new markets – relying on others – is hard to achieve. Moreover, the implantation of key ideas into transformation processes of non-related projects helps to gain additional support in shaping an emerging market without the need for additional pre-commitments.

Paper II implies that practitioners can usually achieve success regardless of their initial network position and shape. However, active optimization of one’s network following the ideas of tertius iungens – by fostering information exchange and collaboration – is essential to improve both success and network position over time. Even if the introduction of non-related
acquaintances deprives entrepreneurs of the options of brokering and arbitrage, they will gain from this form of “intelligent altruism” (Simon, 1990) through co-creation in the long run.

Paper III underlines the importance of increased docility, especially in complex environments. While entrepreneurship literature usually portricts “the entrepreneur” as overly persistent holding out against all struggles (Wood & McKinley, 2010), the results of paper III imply that practitioners need to be extra docile in complex environments. Moreover, they have to make sure that potential stakeholders are sufficiently docile to ensure the successful development of complex opportunities.
References


Dew, N. (2003). Lipsticks and razorblades: how the auto ID center used pre-commitments to build the internet of things. *Available at SSRN 964507*.


Appendices

Appendix A – Overview of simulator structure
AdvancedAnalyses.java

```java
package jEffCauSocialNetworkSimulator;

import java.util.Map;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;
import edu.uci.ics.jung.algorithms.importance.BetweennessCentrality;
import edu.uci.ics.jung.algorithms.metrics.Metrics;
import edu.uci.ics.jung.algorithms.scoring.ClosenessCentrality;
import edu.uci.ics.jung.graph.Graph;
import repast.simphony.context.Context;
import repast.simphony.context.space.graph.ContextJungNetwork;
import repast.simphony.engine.environment.RunEnvironment;
import repast.simphony.engine.schedule.ScheduledMethod;
import repast.simphony.essentials.RepastEssentials;
import repast.simphony.space.graph.Networ
import repast.simphony.space.graph.RepastEdge;

public class AdvancedAnalyses {
    private Network<Object> network;
    private Context<Object> context;
    private int NumberOfIdleTicks = 0;

    public AdvancedAnalyses(Context<Object> _context, Network<Object> _network){
        this.context = _context;
        this.network = _network;
    }

    @ScheduledMethod(start = 0.0, interval = 1, priority = 100)
    public void step(){
        RunEnvironment environment = RunEnvironment.getInstance();
        double tickCount = RepastEssentials.GetTickCount();

        // Network analysis still required?
        if (tickCount > SimulationParameters.Network_analysis_required_until){
            SimulationParameters.Network_analysis_required = false;
        }

        // Is simulation currently stable?
        boolean isSimStatic = true;
        for (Object i : context.getObjects(Agent.class)){
            isSimStatic = isSimStatic && ((Agent)i).isStatic();
        }

        if(!isSimStatic || tickCount< 2.0){
            ExecutorService threadPool = Executors.newFixedThreadPool(SimulationParameters.NumberOfAnalysisThreads);
            if(SimulationParameters.CentralityAnalysisRequired){
                threadPool.submit(new Runnable() {
                    public void run() {
                        calculateBetweennessCentralities();
                    }
                });
            }
        }
    }
}
```
threadPool.submit(new Runnable() {
    public void run()
    {
        getClosenessCentralities();
    }
});
threadPool.submit(new Runnable() {
    public void run()
    {
        getClusteringCoefficients();
    }
});

if(SimulationParameters.StructuralHolesAnalysisRequired){
    StructuralHolesAnalyses.getStructuralHolesMeasures(context, network, threadPool);
}
if(SimulationParameters.GroupCentralityAnalysisRequired){
    GroupCentralityAnalyses.getGroupCentralityMeasures(context, network, threadPool);
}
try {
    threadPool.shutdown();
    threadPool.awaitTermination(Long.MAX_VALUE, TimeUnit.HOURS);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

if (SimulationParameters.MinNumberOfSimSteps > -1){
    if(tickCount <= (double) SimulationParameters.MinNumberOfSimSteps){
        return;
    }
}
if (SimulationParameters.MaxNumberOfSimSteps > -1){
    if(tickCount >= (double) SimulationParameters.MaxNumberOfSimSteps){
        environment.endRun();
    }
}

if (SimulationParameters.doEndIfSimIsStatic == true){
    if(isSimStatic){
        NumberOfIdleTicks++;
    }else{
        NumberOfIdleTicks=0;
    }
    if(NumberOfIdleTicks > SimulationParameters.AcceptableIdleTicks){
        environment.endRun();
    }
}

public int AgentsUsingSpecialFeature(){
    int retVal = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        if (((Effectuator)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
            retVal++;
        }
    }
}
public int EFFsUsingSpecialFeature(){
    int retVal = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        if(i.getClass().getName().equals(Effectuator.class.getName())){
            if(((Effectuator)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
                retVal++;
            }
        }
        return retVal;
    }
}

public int ECsUsingSpecialFeature(){
    int retVal = 0;
    for (Object i: context.getObjects(EndCustomer.class)){
        if(((EndCustomer)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
            retVal++;
        }
    }
    return retVal;
}

public int EFFsUsingSpecialFeature_ByPickup(){
    int retVal = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        if(i.getClass().getName().equals(Effectuator.class.getName())){
            if(((Effectuator)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
                if(((Effectuator)i).hasAdoptedThroughPickup){
                    retVal++;
                }
            }
        }
        return retVal;
    }
}

public int AgentsUsingSpecialFeature_ByNegotiation(){
    int retVal = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        if(((Effectuator)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
            if(((Effectuator)i).hasAdoptedThroughNegotiation){
                retVal++;
            }
        }
    }
    return retVal;
}

public int EFFsUsingSpecialFeature_ByNegotiation(){
    int retVal = 0;
    for (Object i: context.getObjects(Effectuator.class)){

if (i.getClass().getName().equals(Effectuator.class.getName())){
  if (((Effectuator)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
    if (((Effectuator)i).hasAdoptedThroughNegotiation){
      retVal++;
    }
  }
}
return retVal;
}
public int ECsUsingSpecialFeature_ByNegotiation(){
  int retVal = 0;
  for (Object i: context.getObjects(EndCustomer.class)){
    if (((EndCustomer)i).getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
      if (((EndCustomer)i).hasAdoptedThroughNegotiation){
        retVal++;
      }
    }
  }
  return retVal;
}

public int countReasonForNonTransmit_NonPropagation(){
  int retVal = 0;
  for (Object i: context.getObjects(Effectuator.class)){
    retVal+= ((Effectuator)i).countReasonForNonTransmit_NonPropagation;
  }
  return retVal;
}

public int countReasonForNonTransmit_NonRePropagation(){
  int retVal = 0;
  for (Object i: context.getObjects(Effectuator.class)){
    retVal+= ((Effectuator)i).countReasonForNonTransmit_NonRePropagation;
  }
  return retVal;
}

public int countReasonForNonTransmit_NegoFailed_Docility(){
  int retVal = 0;
  for (Object i: context.getObjects(Effectuator.class)){
    retVal+= ((Effectuator)i).countReasonForNonTransmit_NegoFailed_Docility;
  }
  return retVal;
}

public int countReasonForNonTransmit_NegoFailed_Fitness(){
  int retVal = 0;
  for (Object i: context.getObjects(Effectuator.class)){
    retVal+= ((Effectuator)i).countReasonForNonTransmit_NegoFailed_Fitness;
  }
  return retVal;
}

public int countReasonForNonTransmit_SacrInNego(){
  int retVal = 0;
  for (Object i: context.getObjects(Effectuator.class)){
    retVal+= ((Effectuator)i).countReasonForNonTransmit_SacrInNego;
public int countReasonForNonTransmit_NotPickedUp(){
    int retVal = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        retVal += ((Effectuator)i).countReasonForNonTransmit_NotPickedUp;
    }
    return retVal;
}

public double avgGlobalFitness(){
    double retVal = 0;
    double denominator = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        Effectuator EFF =(Effectuator)i;
        retVal += FitnessLandscape.getFitness(EFF.getProductVector());
        denominator++;
    }
    if(denominator>0){
        return retVal/denominator;
    }else{
        return 0;
    }
}

public double avgGlobalFitness_EFFs(){
    double retVal = 0;
    double denominator = 0;
    for (Object i: context.getObjects(Effectuator.class)){
        if(i.getClass().getName().equals(Effectuator.class.getName())){
            Effectuator EFF =(Effectuator)i;
            retVal += FitnessLandscape.getFitness(EFF.getProductVector());
            denominator++;
        }
    }
    if(denominator>0){
        return retVal/denominator;
    }else{
        return 0;
    }
}

public double avgGlobalFitness_ECs(){
    double retVal = 0;
    double denominator = 0;
    for (Object i: context.getObjects(EndCustomer.class)){
        EndCustomer EC =(EndCustomer)i;
        retVal += FitnessLandscape.getFitness(EC.getProductVector());
        denominator++;
    }
    if(denominator>0){
        return retVal/denominator;
    }else{
        return 0;
    }
}

public double avgDeltaFitnessOfSpecialFeatureForECs_ALL(){
    double retVal = 0;
    double denominator = 0;
    for (Object i: context.getObjects(EndCustomer.class)){
        EndCustomer EC =(EndCustomer)i;
        retVal += FitnessLandscape.getFitness(EC.getProductVector());
        denominator++;
    }
    if(denominator>0){
        return retVal/denominator;
    }else{
        return 0;
    }
}
for (Object i: context.getObjects(EndCustomer.class)){
    EndCustomer EC = (EndCustomer)i;
    ProductVector PV_tmp = new ProductVector(EC.getProductVector());
    PV_tmp.setFeature(SimulationParameters.FeatureUnderObservation.getA(), SimulationParameters.FeatureUnderObservation.getB());
    retVal += ((EndCustomer)i).getDeltaFitness(PV_tmp);
    denominator++;
}
if (denominator > 0) {
    return retVal/denominator;
} else {
    return 0;
}
}

class public double avgDeltaFitnessOfSpecialFeatureForECs_WithoutSpecialFeature(){
    double retVal = 0;
    double denominator = 0;
    for (Object i: context.getObjects(EndCustomer.class)){
        EndCustomer EC = (EndCustomer)i;
        if (!EC.getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
            ProductVector PV_tmp = new ProductVector(EC.getProductVector());
            PV_tmp.setFeature(SimulationParameters.FeatureUnderObservation.getA(), SimulationParameters.FeatureUnderObservation.getB());
            retVal += ((EndCustomer)i).getDeltaFitness(PV_tmp);
            denominator++;
        }
    }
    if (denominator > 0) {
        return retVal/denominator;
    } else {
        return 0;
    }
}

class public double getCreatorDegreeCentrality(){
    Effectuator EFF = getCreator();
    if (EFF != null) {
        return EFF.getDegreeCentrality();
    }
    return 0;
}

class public double getCreatorBetweennessCentrality(){
    Effectuator EFF = getCreator();
    if (EFF != null) {
        return EFF.getBetweennessCentrality();
    }
    return 0;
}

class public double getCreatorClosenessCentrality(){
    Effectuator EFF = getCreator();
    if (EFF != null) {
        return EFF.getClosenessCentrality();
    }
    return 0;
}
public double getCreatorClusteringCoefficient()
    { Effectuator EFF = getCreator();
      if (EFF != null)
        return EFF.getClusteringCoefficient();
      return 0;
    }

public double getCreatorDocility()
    { Effectuator EFF = getCreator();
      if (EFF != null)
        return EFF.getDocility();
      return 0;
    }

public double getCreatorProactiveness()
    { Effectuator EFF = getCreator();
      if (EFF != null)
        return (double) EFF.getPropagation_probability();
      return 0;
    }

public double getCreatorMarketFit()
    { Effectuator EFF = getCreator();
      if (EFF != null)
        return (double) EFF.calculateGlobalAverageMarketFit();
      return 0;
    }

private Effectuator getCreator()
    { for (Object i: context.getObjects(Effectuator.class))
        { Effectuator EFF = (Effectuator)i;
          if (EFF.isCPUN_Creator())
            return EFF;
        }
      return null;
    }

private void calculateBetweennessCentralities()
    { if (SimulationParameters.Network_analysis_required)
        { ContextJungNetwork<Object> N = (ContextJungNetwork<Object>)network;
          Graph<Object, RepastEdge<Object>> G = N.getGraph();
          BetweennessCentrality<Object, RepastEdge<Object>> ranker =
            new BetweennessCentrality<Object, RepastEdge<Object>>(G);
          ranker.setRemoveRankScoresOnFinalize(false);
          ranker.evaluate();
          double size = SimulationParameters.NumberOfEffectuators +
            SimulationParameters.NumberOfEndCustomers;
          double NormalizationFactor = (((size-1) * (size-2)) / 2.0);
          double retVal = 0;
          for (Object i : context.getObjects(Effectuator.class))
            { retVal = ranker.getVertexRankScore(i) / NormalizationFactor;
              ((Effectuator)i).setBetweennessCentrality((float)retVal);
            }
        }
    }
private void getClosenessCentralities(){
    if(SimulationParameters.Network_analysis_required){
        ContextJungNetwork<Object> N = (ContextJungNetwork<Object>)network;
        Graph<Object, RepastEdge<Object>> G = N.getGraph();
        ClosenessCentrality<Object, RepastEdge<Object>> ranker = 
            new ClosenessCentrality<Object, RepastEdge<Object>>(G);
        double retVal = 0;
        for (Object i : context.getObjects(Effectuator.class)){
            retVal = ranker.getVertexScore(i);
            ((Effectuator)i).setClosenessCentrality((float)retVal);
        }
    }
}

private void getClusteringCoefficients(){
    if(SimulationParameters.Network_analysis_required){
        Map<Object, Double> cc = 
            Metrics.clusteringCoefficients(((ContextJungNetwork<Object>)network).getGraph());
        double retVal = 0;
        for (Object i : context.getObjects(Effectuator.class)){
            retVal = cc.get(i);
            ((Effectuator)i).setClusteringCoefficient((float)retVal);
        }
    }
}

public int getNumberOfProjects(){
    return Effectuator.projectCounter;
}

Agent.java
package jEffCauSocialNetworkSimulator;

import repast.simphony.context.Context;
import repast.simphony.engine.schedule.ScheduledMethod;
import repast.simphony.space.graph.Network;
import repast.simphony.space.graph.RepastEdge;

public abstract class Agent {
    /**
     * @uml.property name="id"
     */
    private String name = "";
    /**
     * @uml.property name="productVector"
     * @uml.associationEnd inverse="agent: jEffCauSocialNetworkSimulator.ProductVector"
     */
protected ProductVector productVector = new jEffCauSocialNetworkSimulator.ProductVector();

/**
 * @uml.property name="network"
 */
protected Network<Object> network;

/**
 * @uml.property name="context"
 */
protected Context<Object> context;

protected boolean isStatic = true;

/**
 * public Agent(Context<Object> context, Network<Object> network, String id, ProductVector productVector){
 this.setNetwork(network);
 this.setContext(context);
 this.setName(id);
 this.setProductVector(productVector);
 this.setNOTStatic();
}

/**
 * Getter of the property <tt>productVector</tt>
 * @return Returns the productVector.
 * @uml.property name="productVector"
 */
public ProductVector getProductVector() {
 return productVector;
}

/**
 * Setter of the property <tt>productVector</tt>
 * @param productVector The productVector to set.
 * @uml.property name="productVector"
 */
public void setProductVector(ProductVector productVector) {
 this.productVector = productVector;
 setNOTStatic();
}

/**
 * Getter of the property <tt>id</tt>
 * @return Returns the id.
 * @uml.property name="id"
 */
public String getName() {
 return name;
}

/**
 * Setter of the property <tt>id</tt>
 * @param id The id to set.
 * @uml.property name="id"
 */
public void setName(String name) {
 this.name = name;
 setNOTStatic();
}
public Network<Object> getNetwork() {
    return network;
}

public void setNetwork(Network<Object> network) {
    this.network = network;
    setNOTStatic();
}

public Context<Object> getContext() {
    return context;
}

public void setContext(Context<Object> context) {
    this.context = context;
    setNOTStatic();
}

public float getMarketFit(ProductVector testProduct) {
    return 0;
}

public HashMap<ProductVector, Integer> getProductRanking(ArrayList<ProductVector> testProducts) {
    return null;
}

@ScheduledMethod(start = 1, interval = 1, shuffle = true, priority = 300)
public void step() {
    setStatic();
}

public float calculateGlobalAverageMarketFit() {
    assert context != null : "Global market fit cannot be calculated without context!";
    float marketFits = 0F;
    float denominator = 0F;
for (Object ec : context.getObjects(EndCustomer.class)){
    denominator+=1F;
    marketFits += this.getProductVector().calculateFit(((EndCustomer)ec).getProductVector());
}
    assert denominator>0 : "Global market fit cannot be calculated without endCustomers!";
    if(denominator>0.1F){
        return (marketFits)/(denominator);
    }else{
        return 0;
    }
}
public float calculateLocalAverageMarketFit() {
    assert context != null : "Local market fit cannot be calculated without context!";
    assert context != null : "Local market fit cannot be calculated without network!";
    float marketFits = 0F;
    float denominator = 0F;
    for (Object ec : context.getObjects(EndCustomer.class)){
        if(network.getEdge(this, ec) != null){
            denominator+=1F;
            marketFits += this.getProductVector().calculateFit(((EndCustomer)ec).getProductVector());
        }
    }
    //assert denominator>0 : "Global market fit cannot be calculated without (local) endCustomers!";
    if(denominator>0.1F){
        return (marketFits)/(denominator);
    }else{
        return 0;
    }
}
public String getProductVectorString(){
    return getProductVector().toString();
}
public boolean isBusinessPartner(String agentName){
   //checked
   //1. Find Agent by name
   Agent partner = null;
   for (Object i : context.getObjects(Agent.class)){
       Agent curr_agent = (Agent)i;
       if(curr_agent.getName().equals(agentName)){
           partner = curr_agent;
       }
   }

   //2. See if partner is a BusinessPartner (EdgeWeight =2)
   RepastEdge<Object> edge = network.getEdge(this, partner);
   if(edge == null){
       return false;
   }else{
       if(edge.getWeight() == 2){
           return true;
       }else{
           return false;
       }
   }
public boolean isStatic() {
    return this.isStatic;
}

public void setStatic() {
    this.isStatic = true;
}

public void setNOTStatic() {
    this.isStatic = false;
}

BarabasiAlbertNetworkGenerator.java
package jeffCauSocialNetworkSimulator;

import java.util.ArrayList;
import java.util.Iterator;
import repast.simphony.context.Context;
import repast.simphony.context.space.graph.NetworkGenerator;
import repast.simphony.space.graph.Network;

/**
 * @author Jan Willem Jansen
 */
public class BarabasiAlbertNetworkGenerator extends EffectualNetworkGenerator
    implements NetworkGenerator<Object> {

    /**
     * @see repast.simphony.context.space.graph.NetworkGenerator#createNetwork(repast.simphony.space.graph.Network)
     */
    @Override
    public Network<Object> createNetwork(Network<Object> network) {
        return network;
    }

    @Override
    public void step() {
    }

    @Override
    public Network<Object> createIntraAgentNetwork(Network<Object> network, Class<?> targetClass) {
        this.network = network;
    }
}
return createInterAgentNetwork(network, targetClass, targetClass);
}

@Override
public Network<Object> createInterAgentNetwork(Network<Object> network, Class<?> originClass, Class<?> targetClass) {
this.network = network;

//Get all related agents ready
ArrayList<Object> origins = new ArrayList<Object>();
ArrayList<Object> targets = new ArrayList<Object>();

//Get all origins!! Exchange Target-Origins due to specialty of Barabasi-Albert & two-mode-networks
Iterator<Object> iter = context.getObjects(targetClass).iterator();
while (iter.hasNext()){
    Object current = iter.next();
    if (current.getClass().getName().equals(targetClass.getName())){
        origins.add(current);
    }
}

//Get all targets!! Exchange Target-Origins due to specialty of Barabasi-Albert & two-mode-networks
iter = context.getObjects(originClass).iterator();
while (iter.hasNext()){
    Object current = iter.next();
    if (current.getClass().getName().equals(originClass.getName())){
        targets.add(current);
    }
}

for (Object i: origins){
    for (int k = 0; k<SimulationParameters.BAG_EdgesPerStep;k++){
        int[] ConnectionProbabilities = new int[targets.size()];
        for (int j = 0; j < targets.size();j++){
            if (targets.get(j).equals(i)){
                ConnectionProbabilities[j] = 0;
            } else if (network.getEdge(i, targets.get(j)) != null){
                ConnectionProbabilities[j] = 0;
            } else {
                ConnectionProbabilities[j] = 1 + 10*network.getDegree(targets.get(j));
            }
        }
        int draw = RandomHelper.drawfromProbabilityMassFunction(ConnectionProbabilities);
        //Avoid to create Edges twice. Can happen if BAG_EdgesPerStep is too large for reming targets
        if (network.getEdge(i, targets.get(draw)) == null){
            network.addEdge(i, targets.get(draw));
        }
    }
}
return network;
}

@Override
public Network<Object> addCreatorsWithFixedDegrees(Network<Object> network,
    ArrayList<Effectuator> creators, int creatorDegree) {
    for (Effectuator EFF: creators) {
        int degree = 0;
        ArrayList<Effectuator> EFFs = new ArrayList<Effectuator>();
        for (Object i : context.getObjects(Effectuator.class)) {
            EFFS.add((Effectuator) i);
        }
        creatorDegree = Math.min(creatorDegree, EFFs.size());
        while (degree < creatorDegree) {
            int size = EFFS.size();
            int random = Math.abs(RandomHelper.getGenerator().nextInt()) % size;
            Object i = EFFS.get(random);
            if (network.getEdge(EFF, i) == null) {
                network.addEdge(EFF, i);
            }
            EFFS.remove(i);
            degree = network.getDegree(EFF);
        }
    }
    return network;
}

Causator.java
package jEffCauSocialNetworkSimulator;
import java.util.ArrayList;
import repast.simphony.context.Context;
import repast.simphony.space.graph.Network;

/**
 * @author Jan Willem Jansen
 * *
 */
public class Causator extends Agent {

/**
 * *
 */
public Causator(Context<Object> context, Network<Object> network, String id) {
    super(context, network, id, new ProductVector());
}

/**
 * *
 */
public ArrayList<ProductVector> doMarketResearch(Integer
    numberOfParticipants) {
    //checked
    ArrayList<ProductVector> prodVectors = new ArrayList<ProductVector>();
    //Go though all known Contacts
    for (Object cust : context.getRandomObjects(EndCustomer.class,
        numberOfParticipants)) {
        ProductVector prodVec = ((EndCustomer) cust).getCustomerPreferences();
        prodVectors.add(prodVec);
    }
    return prodVectors;
}
public ProductVector createNewOffering(ArrayList<ProductVector> productVectors) {
    //checked
    ProductVector newOffering = new ProductVector();
    int[][] analysis_grid = new int[SimulationParameters.NumberOfFeatures+1][SimulationParameters.NumberOfVariants+1];

    // Create histogram of market research data
    for (ProductVector pv : productVectors) {
        for (int i = 0; i <= SimulationParameters.NumberOfFeatures; i++) {
            analysis_grid[i][pv.getOpinionOn(i)]++;
        }
    }

    // Select highest ranking entries for new Product vector
    for (int feature = 0; feature < analysis_grid.length; feature++) {
        int maxVariant = -1;
        int maxPos = -1;
        for (int value = 0; value < analysis_grid[feature].length; value++) {
            if (analysis_grid[feature][value] > maxVariant) {
                maxVariant = analysis_grid[feature][value];
                maxPos = value;
            }
        }
        newOffering.setFeature(feature, maxPos);
    }
    return newOffering;
}

@Override
public void step() {
    super.step();
    if (SimulationParameters.doMarketResearch) {
        // DO Market research
        ArrayList<ProductVector> marketResearch = doMarketResearch(SimulationParameters.NumberOfMarketResearchParticipants);

        // Create new market offering
        ProductVector newProd = createNewOffering(marketResearch);
        if (!getProductVector().equals(newProd)) {
            setNOTStatic();
        }
        this.setProductVector(newProd);
    }
}

EdgeWeightTransformer.java
package jEffCauSocialNetworkSimulator;

import org.apache.commons.collections15.Transformer;
import repast.simphony.space.graph.RepastEdge;

/**
 * @author Jan Willem Jansen
 */
public class EdgeWeightTransformer implements Transformer<RepastEdge<Object>, Double> {

@Override
public Double transform(RepastEdge<Object> arg0) {
    if (SimulationParameters.fixedEdgeWeight > 0) {
        return Double.valueOf((double) SimulationParameters.fixedEdgeWeight);
    } else {
        return Double.valueOf(arg0.getWeight());
    }
}

EffectualNetworkGenerator.java

package jEffCauSocialNetworkSimulator;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
import bibliothek.util.container.Tuple;
import repast.simphony.context.space.graph.ContextJungNetwork;
import repast.simphony.context.space.graph.NetworkGenerator;
import repast.simphony.context.space.graph.RepastEdge;
import repast.simphony.context.Context;
import repast.simphony.engine.schedule.ScheduledMethod;
import edu.uci.ics.jung.algorithms.metrics.Metrics;
import edu.uci.ics.jung.algorithms.shortestpath.DistanceStatistics;

/**
 * @author Jan Willem Jansen
 * *
 */
public abstract class EffectualNetworkGenerator implements NetworkGenerator<Object> {
    /**
     * *
     */
    public EffectualNetworkGenerator(Context<Object> context) {
        this.context = context;
    }

    /**
     *
     */
    public abstract Network<Object> createNetwork(Network<Object> network);

    /**
     *
     */
    public abstract Network<Object> createIntraAgentNetwork(Network<Object> network, Class<?> targetClass);

    /**
     *
     */
    public abstract Network<Object> createInterAgentNetwork(Network<Object> network, Class<?> originClass, Class<?> targetClass);
@ScheduledMethod(start = 1, interval = 1, shuffle = true, priority = 1)
public abstract void step();

/**
 * @uml.property  name="context"
 */
protected Context<Object> context;
protected Network<Object> network;

/**
 * Getter of the property <tt>context</tt>
 * @return Returns the context.
 * @uml.property  name="context"
 */
public Context<Object> getContext() {
    return context;
}

/**
 * Setter of the property <tt>context</tt>
 * @param context  The context to set.
 * @uml.property  name="context"
 */
public void setContext(Context<Object> context) {
    this.context = context;
}

/**
 * Getter of the property <tt>context</tt>
 * @return Returns the context.
 * @uml.property  name="context"
 */
public Network<Object> getNetwork() {
    return network;
}

/**
 * Setter of the property <tt>context</tt>
 * @param context  The context to set.
 * @uml.property  name="context"
 */
public void setnetwork(Network<Object> network) {
    this.network = network;
}

public Tuple<int[][],ArrayList<String>> getSociomatrix(){
    //checked
    HashMap<Integer, Integer> AgentHashes = new HashMap<Integer, Integer>();
    ArrayList<String> AgentNames = new ArrayList<String>();
    int counter = 0;
    for (Object agent : context.getObjects(Agent.class)) {
        AgentHashes.put(agent.hashCode(), counter);
        AgentNames.add(((Agent)agent).getName());
        counter++;
    }
    int numberOfAgents = AgentHashes.size();
    int[][] Sociomatrix = new int[numberOfAgents][numberOfAgents];
    for (Object ed : network.getEdges()) {
        @SuppressWarnings("unchecked")
        RepastEdge<Object> edge = (RepastEdge<Object>) ed;
int sourceNum = AgentHashes.get(edge.getSource().hashCode());
int targetNum = AgentHashes.get(edge.getTarget().hashCode());
Sociomatrix[sourceNum][targetNum] = (int)(edge.getWeight());
Sociomatrix[targetNum][sourceNum] = (int)(edge.getWeight());
}

Tuple<int[][],ArrayList<String>> retVal = new Tuple<int[][],ArrayList<String>>();
retVal.setA(Sociomatrix);
retVal.setB(AgentNames);
return retVal;

public String getSociomatrixString(){ //checked
    Tuple<int[][],ArrayList<String>> data = getSociomatrix();
    int [][] matrix = data.getA();
    ArrayList<String> headers = data.getB();
    String retVal = "[
    ";
    retVal += Arrays.toString(headers.toArray());
    retVal += ";"
    for (int[] row : matrix){
        retVal += Arrays.toString(row) + ";";
    }
    retVal = retVal.substring(0, retVal.length()-1);
    retVal += "]";
    return retVal;
}

public static double getDensity(Network<Object> network){
    double retVal = ( 2.0 * network.numEdges() ) / ( network.size() * (network.size()-1) );
    return retVal;
}

public static double getDiameter(Network<Object> network){
    double retVal =
    DistanceStatistics.diameter(((ContextJungNetwork<Object>)network).getGraph());
    return retVal;
}

public static double getNetworkClusteringCoefficient(Network<Object> network){
    double retVal = 0;
    Map<Object, Double> cC =
    Metrics.clusteringCoefficients(((ContextJungNetwork<Object>)network).getGraph());
    for (Object n: network.getNodes()) {
        retVal += cC.get(n) / network.size();
    }
    return retVal;
}

public abstract Network<Object>
addCreatorsWithFixedDegrees(Network<Object> network, ArrayList<Effectuator> creators,
    int creatorDegree);
package jEffCauSocialNetworkSimulator;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;

import repast.simphony.context.Context;
import repast.simphony.engine.schedule.ScheduledMethod;
import repast.simphony.essentials.RepastEssentials;
import repast.simphony.space.graph.Network;

/**
 * @author Jan Willem Jansen
 *
 */
public class Effectuator extends Agent {

    protected float docility = 0;
    protected float propagation_probability = 0;
    protected String projectName = "-";
    private ProductVector oldProductVector = new ProductVector();
    public static int projectCounter = 0;
    public boolean isActive = true;

    /**
     * @return the projectCounter
     */
    public static int getProjectCounter() {
        Effectuator.projectCounter++;
        return projectCounter;
    }

    public String getProjectName() {
        return projectName;
    }

    public void setProjectName(String projectName) {
        this.projectName = projectName;
        setNOTStatic();
    }

    public float getPropagation_probability() {
        return propagation_probability;
    }

    public void setPropagation_probability(float propagation_probability) {
        this.propagation_probability = propagation_probability;
        //setNOTStatic();
    }

    protected float distribution_probability = 0;

    /**
     * *
     */
    public Effectuator(Context<Object> context, Network<Object> network, String id,
        ProductVector productVector, float docility, float propagation_probability,

float distribution_probability, boolean isActive) {
    super(context, network, id, productVector);
    setDocility(docility);
    this.propagation_probability = propagation_probability;
    this.distribution_probability = distribution_probability;
    setNOTStatic();
    this.isActive = isActive;
}

@Override
public void step() {
    super.step();
    resetNonTransmitCounters();
    setIncomingIdeasStack();
    // Go through daily routine of an effectuator

    //Think of propagation only if you have sth. new to tell!
    if(!getProductVector().equals(oldProductVector)){
        if(SimulationParameters.doPropagateOwnIdea && this.isActive){ //checked
            //Propagate ideas
            propagateOwnIdea(Effectuator.class);
            propagateOwnIdea(EndCustomer.class);
        }
    }
    oldProductVector = new ProductVector(getProductVector());

    if(SimulationParameters.doPropagateOtherIdeas){ //checked
        //Get Ideas from staple & propagate them
        propagateOtherIdeas();
    }

    if(SimulationParameters.doEvaluatePropagatedIdeas){
        //Evaluate ideas propagated by other effectuators
        evaluatePropagatedIdeas();
    }

    if(SimulationParameters.doPickupNewFeatureFromOtherIdea && this.isActive){
        //Evaluate ideas propagated by other effectuators
        pickupNewFeatureFromOtherIdeas();
    }

    //Finally get rid of new ideas
    incomingIdeas.clear();

    //Do some Self-Measurements
    TotalNumberOfFeatureChanges+=this.getNumberofFeatureChanges();
    checkCPUNassignment();
}

@ScheduledMethod(start = 1, interval = 1, priority = 200)
public void transmitCreatorStrategyToProject(){
    if(isCPU_CREATOR()){
        if(SimulationParameters.doTransmitCreatorStrategyToProject){
            for(Object i : network.getAdjacent(this)){
                if(network.getEdge(this, i).getWeight()==2){
                    ((Effectuator)i).setDocility(this.getDocility());
                    ((Effectuator)i).setPropagation_probability(this.getPropagation_probability());
                }
            }
        }
    }
}
protected void resetNonTransmitCounters() {
    countReasonForNonTransmit_NegoFailed_Docility = 0;
    countReasonForNonTransmit_NegoFailed_Fitness = 0;
    countReasonForNonTransmit_NonPropagation = 0;
    countReasonForNonTransmit_NonRePropagation = 0;
    countReasonForNonTransmit_NotPickedUp = 0;
    countReasonForNonTransmit_SacrInNego = 0;
}

protected void setIncomingIdeasStack() {
    double tickCount = RepastEssentials.GetTickCount();
    incomingIdeas = incomingIdeaStorage.get(tickCount);
    if (incomingIdeas == null) {
        incomingIdeas = new HashMap<ProductVector, ArrayList<String>>();
    }
}

protected void pickupNewFeatureFromOtherIdeas() {
    for (ProductVector pv : incomingIdeas.keySet()) {
        // due to asynchronous processing, we have to exclude ideas from new
        // found business partners
        // due to repropagation of our idea, we have to make sure to exclude our
        // own ideas
        String sender = incomingIdeas.get(pv).get(0);
        if (!sender.equals(this.getName()) && !isBusinessPartner(sender)) {
            if (SimulationParameters.doPickupNewFeatureFromOtherIdea) {
                if (RandomHelper.getGenerator().nextDouble() <= this.getDocility()) {
                    ProductVector newIdeas = pickupNewFeatureFromOtherIdea(pv);
                    if (!newIdeas.isEmpty()) {
                        ProductVector copy = new ProductVector(getProductVector());
                        getProductVector().incorporateIdeas(newIdeas);
                        setNOTStatic();
                        // Check if this pickup led to (non-)incorporation of special
                        feature
                        checkHasAdoptedThroughPickup();
                        if (!getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation) &&
                            pv.hasFeature(SimulationParameters.FeatureUnderObservation)) {
                            countReasonForNonTransmit_NotPickedUp++;
                        }
                        // Negotiate inclusion with my business partners
                        Negotiation myNego = new ProbabilisticNegotiation(context,
                            network);
                        myNego.setPartyA(this);
                        myNego.setPartyB(this);
                        boolean result = myNego.negotiate();
                        if (!result) {
                            assert 1 < 0 : "That should not have happened! BUG???";
                            setProductVector(copy);
                        }
                    }
                } else {
                    // Measurement: see if low docility caused Non-Pickup of "special
                    feature"
                }
            } else {
                // Measurement: see if low docility caused Non-Pickup of "special
                feature"
            }
        } else {
        }
    }
}
protected void propagate(ProductVector idea, Class<?> targetclass, ArrayList<String> ids){ //checked
  //Go through all known Contacts
  for (Object neighbor: network.getAdjacent(this)) {
    //identify targets
    if (neighbor.getClass().getName().equals(targetclass.getName())) {
      //If this guy is not already with you
      if (network.getEdge(this, neighbor).getWeight() != 2) {
        //Pitch your idea if you feel like it
        Effectuator EFF = (Effectuator) neighbor;
        Random generator = RandomHelper.getGenerator();
        if (generator.nextFloat() <= this.propagation_probability) {
          EFF.addIncomingIdea(idea, ids);
        }
      }
    } else {
      if (getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)) {
        CPUN_Propagator = true;
      } else {
        if (getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)) {
          if (ids.get(0).equals(getName())) {
            countReasonForNonTransmit_NonPropagation++;
          } else {
            countReasonForNonTransmit_NonRePropagation++;
          }
        }
      }
    } else {
      if (getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)) {
        countReasonForNonTransmit_NotPickedUp++;
      }
    }
  }
}

protected ProductVector pickupNewFeatureFromOtherIdea(ProductVector otherIdea){ //checked
  ProductVector newIdea = new ProductVector();
  ArrayList<Integer> newfeatures = otherIdea.getKnownFeatures();
  newfeatures.removeAll(getProductVector().getKnownFeatures());
  if (newfeatures.size() > 0) {
    int position = RandomHelper.getGenerator().nextInt(newfeatures.size());
    newIdea.setFeature(newfeatures.get(position), otherIdea.getOpinionOn(newfeatures.get(position)));
  }
  return newIdea;
}

protected void propagateOwnIdea(Class<?> clazz){ //checked
ArrayList<String> ids = new ArrayList<String>();
ids.add(getName());
propagate(getProductVector(), clazz, ids);
}

protected void propagateOtherIdeas(){
    for (ProductVector pv: incomingIdeas.keySet()){
        //if(RandomHelper.getGenerator().nextFloat() <=
        this.distribution_probability){
            ArrayList<String> ids = incomingIdeas.get(pv);
            ids.add(getName());
            propagate(pv, Effectuator.class, ids);
            propagate(pv, EndCustomer.class, ids);
        //}
    }
}

protected HashMap<Double, HashMap<ProductVector, ArrayList<String>>> incomingIdeaStorage = new HashMap<Double,
HashMap<ProductVector,ArrayList<String>>>();
protected HashMap<ProductVector, ArrayList<String>> incomingIdeas = null;

protected void evaluatePropagatedIdeas(){
    ArrayList<String> removeLater = new ArrayList<String>();
    for (ProductVector pv : incomingIdeas.keySet()){
        String sender = incomingIdeas.get(pv).get(0);
        //due to asynchronous processing, we have to exclude ideas from new
        //found business partners
        //due to repropagation of our idea, we have to make sure to exclude our
        own ideas
        if(!sender.equals(this.getName()) && !isBusinessPartner(sender)){
            Negotiation myNegotiation = new ProbabilisticNegotiation(context,
network);
            boolean hasSuccessfullyNegotiated = executeNegotiation(sender,
myNegotiation);
            if(hasSuccessfullyNegotiated){
                setNOTStatic();
                //See if negotiation led to Pickup of special feature or its loss in
                the negotiation
                checkHasAdoptedOrLostThroughNegotiation();
                removeLater.add(sender);
            }
        }else{
            //See if negotiation failed and let to non-propagation of special
            feature
            if(!getProductVector().hasFeature(SimulationParameters.FeatureUnderObserva
tion) &&
                pv.hasFeature(SimulationParameters.FeatureUnderObservation)){
                if(myNegotiation.reasonForFailure_Docility){
                    countReasonForNonTransmit_NegoFailed_Docility++;
                }
                if(myNegotiation.reasonForFailure_Fitness){
                    countReasonForNonTransmit_NegoFailed_Fitness++;
                }
            }
        }
    }
    //Clear List of successfully negotiated Ideas for Pickup-Stuff
    for (String i : removeLater){
        incomingIdeas.remove(i);
    }
}
public boolean evaluatePropagatedIdea(String AgentID, ProductVector idea) {
  //checked
  //if (this.getProductVector().isAcceptable(idea, getDocility())){
  //  return executeNegotiation(AgentID);
  //}
  //return false;
}

protected boolean executeNegotiation(String AgentID, Negotiation myNegotiation) {
  //checked

  myNegotiation = new ProbabilisticNegotiation(context, network);
  for (Object agent : context.getObjects(Effectuator.class)) {
    if (((Agent)agent).getName().equals(AgentID)) {
      myNegotiation.setPartyA((Effectuator)agent);
    }
  }
  myNegotiation.setPartyB(this);
  return myNegotiation.negotiate();
}

public boolean addIncomingIdea(ProductVector incomingIdea, ArrayList<String> ids) {
  double tickCount = RepastEssentials.GetTickCount();
  if (incomingIdeaStorage.get(tickCount + 1) == null) {
    incomingIdeaStorage.put(tickCount + 1, new HashMap<ProductVector, ArrayList<String>>());
  }
  HashMap<ProductVector, ArrayList<String>> FutureIncomingIdeas = incomingIdeaStorage.get(tickCount + 1);
  if (!ids.contains(getName())) {
    FutureIncomingIdeas.put(incomingIdea, ids);
    return true;
  } else {
    return false;
  }
}

/**
 * Getter of the property <tt>docility</tt>
 * @return Returns the docility.
 * @uml.property  name="docility"
 */
public float getDocility() {
  return docility;
}

/**
 * Setter of the property <tt>docility</tt>
 * @param docility The docility to set.
 * @uml.property  name="docility"
 */
public void setDocility(float docility) {
  this.docility = docility;
  //setNOTStatic();
}

public boolean isNegotiationResultAcceptable(ProductVector result) {
}
return getProductVector().isAcceptable(result, getDocility());
}

protected boolean hasAdoptedThroughPickup = false;
protected boolean hasAdoptedThroughNegotiation = false;

public int countReasonForNonTransmit_NonPropagation = 0;
public int countReasonForNonTransmit_NonRePropagation = 0;
public int countReasonForNonTransmit_NegoFailed_Docility = 0;
public int countReasonForNonTransmit_NegoFailed_Fitness = 0;
public int countReasonForNonTransmit_SacrInNego = 0;
public int countReasonForNonTransmit_NotPickedUp = 0;

public void checkHasAdoptedThroughPickup(){
if(getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
    if(hasAdoptedThroughPickup || hasAdoptedThroughNegotiation){
        return;
    }
    else{
        hasAdoptedThroughPickup = true;
    }
    hasAdoptedThroughPickup = false;
    hasAdoptedThroughNegotiation = false;
}
}

public void checkHasAdoptedOrLostThroughNegotiation(){
if(getProductVector().hasFeature(SimulationParameters.FeatureUnderObservation)){
    if(hasAdoptedThroughPickup || hasAdoptedThroughNegotiation){
        return;
    }
    else{
        hasAdoptedThroughNegotiation = true;
    }
    if(hasAdoptedThroughPickup || hasAdoptedThroughNegotiation){
        countReasonForNonTransmit_SacrInNego++;
    }
    hasAdoptedThroughPickup = false;
    hasAdoptedThroughNegotiation = false;
}
//Creator, User, Propagator, None of the above - Einteilung:
public boolean CPUN_None = false;
public boolean CPUN_Creator = false;
public boolean CPUN_Propagator = false;
public boolean CPUN_User = false;

public boolean isCPUN_None() {
    return CPUN_None;
}

public boolean isCPUN_Creator() {
    return CPUN_Creator;
}

public boolean isCPUN_Propagator() {
    return CPUN_Propagator;
}

public boolean isCPUN_User() {
    return CPUN_User;
}

protected void checkCPUNassignment(){
    if (hasAdoptedThroughPickup || hasAdoptedThroughNegotiation ||
        CPUN_Creator){
        CPUN_User = true;
    }else{
        CPUN_User = false;
    }
    this.CPUN_None = !(CPUN_Creator && CPUN_Propagator && CPUN_User);
}

public void setCreator(boolean isCreator) {
    this.CPUN_Creator = isCreator;
}

public double getAbsoluteFitness(ProductVector PV){
    return FitnessLandscape.getFitness(PV);
}

public float getEFF_Fitness(){
    return (float) FitnessLandscape.getFitness(getProductVector());
}

protected float betweennessCentrality = 0.0f;
protected float closenessCentrality = 0.0f;
protected float clusteringCoefficient = 0.0f;
protected float groupDegreeCentrality = 0.0f;
protected float groupBetweennessCentrality = 0.0f;
protected float groupClosenessCentrality = 0.0f;
protected float constraint = 0.0f;
protected float effectiveSize = 0.0f;
protected float efficiencyStructHoles = 0.0f;
protected float hierarchy = 0.0f;

public double getDegreeCentrality(){
double retVal = (double)network.getDegree(this) / (double)(SimulationParameters.NumberOfEffectuators + SimulationParameters.NumberOfEndCustomers);
return retVal;
}

public float getBetweennessCentrality() {
    return betweennessCentrality;
}

public void setBetweennessCentrality(float betweennessCentrality) {
    this.betweennessCentrality = betweennessCentrality;
}

public float getClosenessCentrality() {
    return closenessCentrality;
}

public void setClosenessCentrality(float closenessCentrality) {
    this.closenessCentrality = closenessCentrality;
}

/**
 * @return the clusteringCoefficient
 */
public float getClusteringCoefficient() {
    return clusteringCoefficient;
}

/**
 * @param clusteringCoefficient the clusteringCoefficient to set
 */
public void setClusteringCoefficient(float clusteringCoefficient) {
    this.clusteringCoefficient = clusteringCoefficient;
}

/**
 * @return the groupDegreeCentrality
 */
public float getGroupDegreeCentrality() {
    return groupDegreeCentrality;
}

/**
 * @param groupDegreeCentrality the groupDegreeCentrality to set
 */
public void setGroupDegreeCentrality(float groupDegreeCentrality) {
    this.groupDegreeCentrality = groupDegreeCentrality;
}

/**
 * @return the groupBetweennessCentrality
 */
public float getGroupBetweennessCentrality() {
    return groupBetweennessCentrality;
}

/**
 * @param groupBetweennessCentrality the groupBetweennessCentrality to set
 */
public void setGroupBetweennessCentrality(float groupBetweennessCentrality) {
this.groupBetweennessCentrality = groupBetweennessCentrality;
}

/**
 * @return the groupClosenessCentrality
 */
public float getGroupClosenessCentrality() {
    return groupClosenessCentrality;
}

/**
 * @param groupClosenessCentrality the groupClosenessCentrality to set
 */
public void setGroupClosenessCentrality(float groupClosenessCentrality) {
    this.groupClosenessCentrality = groupClosenessCentrality;
}

/**
 * @return the constraint
 */
public float getConstraint() {
    return constraint;
}

/**
 * @param constraint the constraint to set
 */
public void setConstraint(float constraint) {
    this.constraint = constraint;
}

/**
 * @return the effectiveSize
 */
public float getEffectiveSize() {
    return effectiveSize;
}

/**
 * @param effectiveSize the effectiveSize to set
 */
public void setEffectiveSize(float effectiveSize) {
    this.effectiveSize = effectiveSize;
}

/**
 * @return the efficiency
 */
public float getEfficiencyStructHoles() {
    return efficiencyStructHoles;
}

/**
 * @param efficiency the efficiency to set
 */
public void setEfficiencyStructHoles(float efficiencyStructHoles) {
    this.efficiencyStructHoles = efficiencyStructHoles;
}

/**
 * @return the hierarchy
 */
public float getHierarchy() {
    return hierarchy;
}

/**
 * @param hierarchy the hierarchy to set
 */
public void setHierarchy(float hierarchy) {
    this.hierarchy = hierarchy;
}

public int getProjectSize() {
    // return project size
    int retVal = 1;
    if (this.getProjectName().equals("-")){
        return retVal;
    }
    for (Object i : network.getAdjacent(this)){
        if ( ((Effectuator) i).getProjectName().equals(this.getProjectName())){
            retVal++;
        }
    }
    return retVal;
}

private int oldNumberOfStakeholders = 0;
public int getNumberOfStakeholders(){
    return getProjectSize()-1;
}

public int getNumberOfNewStakeholders(){
    int retVal = getNumberOfStakeholders() - oldNumberOfStakeholders;
    oldNumberOfStakeholders = getNumberOfStakeholders();
    return retVal;
}

public int getNumberOfFeatureChanges(){
    if(oldProductVector.isEmpty()){return 0;
    }else{
        return getProductVector().getNumberOfFeatureChanges(oldProductVector);
    }
}

private int TotalNumberOfFeatureChanges=0;
public int getTotalNumberOfFeatureChanges(){
    return TotalNumberOfFeatureChanges;
}

public int getNumberOfNewFeatures(){
    if(oldProductVector.isEmpty()){return 0;
    }else{
        return getProductVector().getNumberOfNewFeatures(oldProductVector);
    }
}

public int getAbsoluteNumberOfFeatures(){
    return getProductVector().getKnownFeatures().size();}
private int initialPVType = -1;

public int getInitialPVType() {
    return initialPVType;
}

public void setInitialPVType(int initialPVType) {
    this.initialPVType = initialPVType;
}

private ArrayList<Integer> getStakeholderInitialPVTypeDistribution_list(){
    ArrayList<Integer> PVTypeDistribution_list = new ArrayList<Integer>();
    if(SimulationParameters.PVGeneratorType.equals("Predef")){
        //PrepareList
        for (int i = 0; i< SimulationParameters.PVPathDependency_NumberOfTypes; i++){
            PVTypeDistribution_list.add(0);
        }
        //Add yourself
        PVTypeDistribution_list.set(getInitialPVType()-1, 1);
        //Add the others
        for(Object i : network.getAdjacent(this)){
            if(network.getEdge(this, i).getWeight() > 1){
                Effectuator EFF = (Effectuator) i;
                if(EFF.getInitialPVType()>0){
                    int count = PVTypeDistribution_list.get(EFF.getInitialPVType()-1);
                    PVTypeDistribution_list.set(EFF.getInitialPVType()-1, count+1);
                }
            }
        }
        return PVTypeDistribution_list;
    }
    return new ArrayList<Integer>();
}

public String getStakeholderInitialPVTypeDistribution_exact(){
    String retVal = "",
    if(SimulationParameters.PVGeneratorType.equals("Predef")){
        ArrayList<Integer> InitPVDist = getStakeholderInitialPVTypeDistribution_list();
        //Calculate total number of stakeholders
        float sum = 0;
        for(Integer i: InitPVDist){
            sum +=i;
        }
        //Calculate percentages
        for(Integer i: InitPVDist){
            retVal += Float.toString((float)i / sum )+",";
        }
        return retVal;
    }
}

public String getStakeholderInitialPVTypeDistribution_rounded(){
    String retVal = "",
    if(!SimulationParameters.PVGeneratorType.equals("-")){
ArrayList<Integer> InitPVDist =
getStakeholderInitialPVTypeDistribution_list();

    //Calculate total number of stakeholders
    float sum = 0;
    for (Integer i: InitPVDist){
        sum += i;
    }

    //Calculate percentages
    float factor = 5;
    for (Integer i: InitPVDist){
        float val = Math.round(i*factor / (sum)) / factor;
        retVal += Float.toString(val)+",";
    }
    return retVal;
}
```java
@Override
public boolean isNegotiationResultAcceptable(ProductVector result){
    if(FitnessLandscape.getFitness(result) >= FitnessLandscape.getFitness(getProductVector())){
        return true;
    }else{
        return false;
    }
}

public double getDeltaFitness(ProductVector PV){
    return FitnessLandscape.getFitness(PV) - FitnessLandscape.getFitness(getProductVector());
}

package jEffCauSocialNetworkSimulator;

public class FitnessLandscape {

    protected static double[][] staticFitnessValues;

    public FitnessLandscape() {
        staticFitnessValues = new double[0][0];
    }

    public FitnessLandscape(int NumberOfFeatures, int NumberOfValues) {
        staticFitnessValues = new double[NumberOfFeatures][NumberOfValues+1];
    }

    public void createRandomFitnesslandscape(double minVal, double maxVal){
        for (int i = 0; i<staticFitnessValues.length; i++){
            for (int j = 0; j < staticFitnessValues[0].length; j++){
                staticFitnessValues[i][j] = RandomHelper.getGenerator().nextDouble() * (maxVal - minVal) + minVal;
            }
        }
    }

    public static double getFitness(ProductVector request){
        double retVal = 0;
        for (Integer i : request.getKnownFeatures()){
            for (int j = 0; j<staticFitnessValues.length; j++){
                staticFitnessValues[i-1][j] = staticFitnessValues[i][j-1] = staticFitnessValues[i][j] = 0;
            }
        }
    }

    public static double getFitness(ProductVector request){
        double retVal = 0;
        for (Integer i : request.getKnownFeatures()){
            retVal += staticFitnessValues[i-1][request.getOpinionOn(i)];
        }
        for (int i = 0; i<staticFitnessValues.length;i++){
            retVal += staticFitnessValues[i][request.getOpinionOn(i+1)];
        }
        return retVal;
    }

    public static void clearFeature(int feature){
        for (int i = 0; i<staticFitnessValues[feature-1].length;i++){
            staticFitnessValues[feature-1][i]=0;
        }
    }
}
```
public static void setFeatureVariant(int feature, int variant, double value) {
    staticFitnessValues[feature-1][variant]=value;
}

package jEffCauSocialNetworkSimulator;

import repast.simphony.space.graph.Network;
import repast.simphony.context.Context;
import java.util.ArrayList;

/**
* @author Jan Willem Jansen
*
*/
public abstract class Negotiation {

    protected Effectuator[] Effectuators = {null, null};
    protected Context<Object> context = null;
    protected Network<Object> network = null;

    protected boolean reasonForFailure_Docility = false;
    protected boolean reasonForFailure_Fitness = false;

    public Negotiation(Context<Object> context, Network<Object> network) {
        this.context = context;
        this.network = network;
    }

    public Negotiation(Context<Object> context, Network<Object> network, Effectuator PartyA, Effectuator PartyB) {
        this(context, network);
        setPartyA(PartyA);
        setPartyB(PartyB);
    }

    public void setPartyA(Effectuator PartyA) {
        Effectuators[0] = PartyA;
    }

    public Effectuator getPartyA() {
        return Effectuators[0];
    }

    public void setPartyB(Effectuator PartyB) {
        Effectuators[1] = PartyB;
    }

    public Effectuator getPartyB() {
        return Effectuators[0];
    }

    abstract public boolean negotiate();
abstract protected ProductVector calculateResult(ArrayList<ProductVector> interestsA, ArrayList<ProductVector> interestsB);

protected void createEdgesfromAtoB(ArrayList<Effectuator> involvedAgentsA, ArrayList<Effectuator> involvedAgentsB, double EdgeWeight) {
    //Cycle through parties
    for (Effectuator agentA : involvedAgentsA) {
        for (Effectuator agentB : involvedAgentsB) {
            assert !agentA.equals(agentB) : "Negotiation let to creation of loop in network from agentA to agentA";
            NetworkEdge myEdge = (NetworkEdge) network.getEdge(agentA, agentB);
            //Create/Update network of weight EdgeWeight
            if (myEdge == null) {
                myEdge = (NetworkEdge) network.addEdge(agentA, agentB, EdgeWeight);
            } else {
                myEdge.setWeight(Math.max(EdgeWeight, myEdge.getWeight()));
            }
        }
    }
}

protected ArrayList<Effectuator> gatherInvolvedAgents(Effectuator party) {
    //checked
    ArrayList<Effectuator> involvedAgents = new ArrayList<Effectuator>();
    involvedAgents.add(party);
    for (Object i : network.getAdjacent(party)) {
        if (network.getEdge(party, i).getWeight() == 2) {
            involvedAgents.add((Effectuator)i);
        }
    }
    return involvedAgents;
}

protected ArrayList<ProductVector> gatherInterests(ArrayList<Effectuator> involvedAgents) {
    //checked
    ArrayList<ProductVector> interests = new ArrayList<ProductVector>();
    for (Effectuator EFF : involvedAgents) {
        interests.add(EFF.getProductVector());
    }
    return interests;
}

NetworkEdge.java
package jEffCauSocialNetworkSimulator;

import java.awt.Color;
import java.util.ArrayList;
import repast.simphony.space.graph.RepastEdge;

/**
 * @author willem
 *
 */
public class NetworkEdge extends RepastEdge<Object> {
    private double thickness;
    private Color color;
    private ArrayList<Color> colorList = new ArrayList<Color>();

    public NetworkEdge() {
        this(null, null, false, 1);
    }
public NetworkEdge(Object source, Object target, boolean directed) {
    this(source, target, directed, 1);
}

public NetworkEdge(Object source, Object target, boolean directed, double weight) {
    super(source, target, directed, weight);
    colorList.add(Color.GREEN);
    colorList.add(Color.RED);
    colorList.add(Color.BLUE);
    colorList.add(Color.ORANGE);
    colorList.add(Color.CYAN);
    colorList.add(Color.PINK);
    colorList.add(Color.YELLOW);
    colorList.add(Color.GRAY);
    colorList.add(Color.LIGHT_GRAY);
    colorList.add(Color.MAGENTA);
    setWeight(weight);
}

@Override
public void setWeight(double weight){
    super.setWeight(weight);
    if(weight == 1){
        setColor(Color.BLACK);
        setThickness(20.0);
    } else if(weight == 2){
        String projectName = ((Effectuator) this.getSource()).getProjectName();
        int projectNumber = Integer.parseInt(projectName.split(“_”)[1]);
        setColor(colorList.get(projectNumber % colorList.size()));
        setThickness(35.0);
    } else{
        // so far undefined!
        setColor(Color.RED);
        setThickness(1.0);
    }
}

/**
 * @return the thickness
 */
public double getThickness() {
    double retVal = thickness;
    return retVal;
}

/**
 * @param thickness the thickness to set
 */
public void setThickness(double thickness) {
    this.thickness = thickness;
}

/**
 * @return the color
 */
public Color getColor() {
    return color;
}

/**
public void setColor(Color color) {
    this.color = color;
}

public double getRed() {
    double retVal = color.getRed();
    return retVal;
}

public double getGreen() {
    double retVal = color.getGreen();
    return retVal;
}

public double getBlue() {
    double retVal = color.getBlue();
    return retVal;
}

public void resetVisualization(){
    setWeight(getWeight());
}

ProbabilisticNegotiation.java
package jEffCauSocialNetworkSimulator;
import java.util.ArrayList;
import repast.simphony.context.Context;
import repast.simphony.space.graph.Network;

/**
 * @author Jan Willem Jansen
 * *
 */
public class ProbabilisticNegotiation extends Negotiation {

    /**
     * *
     */
    public ProbabilisticNegotiation(Context<Object> context, Network<Object> network) {
        super(context, network);
    }

    public ProbabilisticNegotiation(Context<Object> context, Network<Object> network, Effectuator PartyA, Effectuator PartyB) {
        super(context, network, PartyA, PartyB);
    }

    @Override
    public boolean negotiate() { //checked
        ArrayList<Effectuator> involvedAgentsA = gatherInvolvedAgents(Effectuators[0]);
        ArrayList<Effectuator> involvedAgentsB = gatherInvolvedAgents(Effectuators[1]);
        involvedAgentsB.removeAll(involvedAgentsA);
        ArrayList<ProductVector> interestsA = gatherInterests(involvedAgentsA);
        ArrayList<ProductVector> interestsB = gatherInterests(involvedAgentsB);
        return interestsA.containsAll(interestsB);
ProductVector result = calculateResult(interestsA, interestsB);
boolean involvedAgentsA_agree = true;
boolean involvedAgentsB_agree = true;
for (Effectuator i : involvedAgentsA){
    boolean answer = i.isNegotiationResultAcceptable(result);
    involvedAgentsA_agree = involvedAgentsA_agree && answer;
    if (!answer){
        if (i.getClass().getName().equals(Effectuator.class.getName())){
            this.reasonForFailure_Docility = true;
        } else {
            this.reasonForFailure_Fitness = true;
        }
    }
}
for (Effectuator i : involvedAgentsB){
    boolean answer = i.isNegotiationResultAcceptable(result);
    involvedAgentsB_agree = involvedAgentsB_agree && answer;
    if (!answer){
        if (i.getClass().getName().equals(Effectuator.class.getName())){
            this.reasonForFailure_Docility = true;
        } else {
            this.reasonForFailure_Fitness = true;
        }
    }
}

// If both parties agree, connect them and create joint project
if (involvedAgentsA_agree && involvedAgentsB_agree){
    String ProjectName = "";
    if (Effectuators[0].getProjectName().equals("-")){
        if (Effectuators[1].getProjectName().equals("-")){
            ProjectName = "Proj" + Integer.toString(Effectuator.getProjectCounter());
        } else {
            ProjectName = Effectuators[1].getProjectName();
        }
    } else {
        ProjectName = Effectuators[0].getProjectName();
    }
    for (Effectuator i : involvedAgentsA){
        i.setProductVector(result);
        i.setProjectName(ProjectName);
        i.checkHasAdoptedOrLostThroughNegotiation();
        i.setNOTStatic();
        i.isActive = true;
    }
    for (Effectuator i : involvedAgentsB){
        i.setProductVector(result);
        i.setProjectName(ProjectName);
        i.checkHasAdoptedOrLostThroughNegotiation();
        i.setNOTStatic();
        i.isActive = true;
    }
    createEdgesfromAtoB(involvedAgentsA, involvedAgentsB, 2);
    return true;
} else {
    //createEdgesfromAtoB(involvedAgentsA, involvedAgentsB, 1);
    return false;
}
@Override
protected ProductVector calculateResult(ArrayList<ProductVector> interestsA, ArrayList<ProductVector> interestsB) { //checked
    ProductVector retval = new ProductVector();
    // Create histogram array
    int[][] histogram = new int[SimulationParameters.NumberOfFeatures+1][SimulationParameters.NumberOfVariants+1];
    // Populate with preferences of Parties A & B
    for (ProductVector PV : interestsA){
        for (int feature: PV.getKnownFeatures()){
            histogram[feature][PV.getOpinionOn(feature)] += 1;
        }
    }
    for (ProductVector PV : interestsB){
        for (int feature: PV.getKnownFeatures()){
            histogram[feature][PV.getOpinionOn(feature)] += 1;
        }
    }
    // Clear "0" = unknown feature column to avoid "majority vote" on exclusion of a feature
    for (int[] feature: histogram){ feature[0] = 0; }
    // Create result of negotiation based on probabilistic negotiation for required features
    for (int i = 1; i < histogram.length; i++){
        // See if line in histogram was used aka. feature is really known to anyone
        int sum = 0;
        for (int j : histogram[i]){
            sum += j;
        }
        if(sum > 0){
            Integer drawResult = RandomHelper.drawfromProbabilityMassFunction(histogram[i]);
            retval.setFeature(i, drawResult);
        }
    }
    return retval;
}

ProductVector.java
package jEffCauSocialNetworkSimulator;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;
import bibliothek.util.container.Tuple;

/**
 * @author Jan Willem Jansen
 */
public class ProductVector {
    protected HashMap<Integer, Integer> prodvector = new HashMap<Integer, Integer>();
    /**
     * 
     */
public ProductVector() {
}

public ProductVector(ProductVector originalProductVector) {
    for (Integer feature : originalProductVector.getKnownFeatures()){
        this.setFeature(feature, originalProductVector.getOpinionOn(feature));
    }
}

/**
 * Creates randomly assigned Productvector with NumOfFeatures elements based on FeatureRange and ValueRange
 */
public ProductVector(Integer FeatureRange, Integer ValueRange, Integer NumOfFeatures, boolean isEndCustomer) {
    Random generator = RandomHelper.getGenerator();
    while (prodvector.size() < NumOfFeatures){
        Integer randFeature = generator.nextInt(FeatureRange)+1;
        Integer randValue = 0;
        if(!isEndCustomer){
            randValue = generator.nextInt(ValueRange)+1;
        }else{
            double[] fitness = FitnessLandscape.staticFitnessValues[randFeature-1];
            int[] int_fitness = new int[fitness.length];
            for(int i=0; i<fitness.length;i++){
                int_fitness[i] = (int)(10000*fitness[i]);
            }
            randValue = RandomHelper.drawfromProbabilityMassFunction(int_fitness);
        }
        setFeature(randFeature, randValue);
    }
}

/**
 * Getter of the property <tt>prodvector</tt>
 * @return Returns the prodvector.
 * @uml.property name="prodvector"
 */
HashMap<Integer, Integer> getProdvector() {
    return prodvector;
}

/**
 * Setter of the property <tt>Pvector</tt>
 * @param Pvector The pvector to set.
 * @uml.property name="Pvector"
 */
public void setProdvector(HashMap<Integer, Integer> prodvector) {
    this.prodvector = prodvector;
}

/**
 * Returns Opinion on a given feature. '0' indicates "unknown Feature"
 */
public Integer getOpinionOn(Integer feature){
    if (prodvector.containsKey(feature)){
        return prodvector.get(feature);
    }else{
public ArrayList<Integer> getKnownFeatures(){
    return new ArrayList<Integer>(prodvector.keySet());
}

/**
 * Sets Opinion on a given feature. '0' indicates "unknown Feature/No
 * Opinion" and deletes it. Feature '0' is ignored)
 */
public Boolean setFeature(Integer feature, Integer value){
    if (feature != 0){
        if (value != 0){
            prodvector.put(feature, value);
        }else{
            prodvector.remove(feature);
        }
        return true;
    }else{
        return true;
    }
}

public Boolean isAcceptable(ProductVector otherVector, float docility){
    //checked
    if (requiredDocility(otherVector) > docility){
        return false;
    }else{
        return true;
    }
}

public float requiredDocility(ProductVector otherVector){
    //checked
    int positives = 0;
    int negatives = 0;
    int no_overlap = 0; //Count occurrences of features

    ArrayList<Integer> knownFeatures = this.getKnownFeatures(); //Get all Known Features
    for (Integer i : knownFeatures){
        //Ignore the ones we have no opinion on
        if (getOpinionOn(i) != 0){
            //Count if other Vector does not interfere
            if (otherVector.getOpinionOn(i) == 0){
                no_overlap++;
            }else{
                if (this.getOpinionOn(i) == otherVector.getOpinionOn(i)){
                    //Count as positives if the opinion is the same
                    positives++;
                }else{
                    //Count as negatives if opinion differs
                    negatives++;
                }
            }
        }
    }
    // Return relative amount of negatives over all my features
    return (float) negatives / (float)(negatives+positives+no_overlap);
}
@Override
public boolean equals(Object otherVector){
    if(otherVector instanceof ProductVector){
        ProductVector otherPV = (ProductVector) otherVector;
        if(getKnownFeatures().size() != otherPV.getKnownFeatures().size()){
            return false;
        }
        for(int i : getKnownFeatures()){  
            if(this.getOpinionOn(i) != otherPV.getOpinionOn(i)){
                return false;
            }
        }
        return true;
    }else{
        return false;
    }
}

public boolean incorporateIdeas(ProductVector newIdea){
    for (Integer feature : newIdea.getKnownFeatures()){  
        if(this.getOpinionOn(feature) != 0){
            return false;
        }
    }
    for (Integer newFeature : newIdea.getKnownFeatures()){  
        this.setFeature(newFeature, newIdea.getOpinionOn(newFeature));
    }
    return true;
}

public int size(){
    return prodvector.size();
}

public float calculateFit(ProductVector customerIdea){
    //Calculate market fit for 2 Product Vectors
    //Idea: Fit = (#features compliant with customer needs) / total number
    //of customer’s features
    int denominator = customerIdea.size();
    int compliantFeatures = 0;
    for (int feature : customerIdea.getKnownFeatures()){  
        if(customerIdea.getOpinionOn(feature) == this.getOpinionOn(feature)){
            compliantFeatures++;
        }
    }
    float retVal = ((float)compliantFeatures) / ((float) denominator);
    return retVal;
}

@Override
public String toString(){ //checked 
    String retVal = "\n";
    for (int i=1; i<=SimulationParameters.NumberOfFeatures; i++){  
        if(getOpinionOn(i) != 0){
            retVal += "" + Integer.toString(getOpinionOn(i)) + ",";
        }else{
            retVal += "-" + ",";
        }
    }
    retVal = retVal.substring(0, retVal.length() -1);
}
public boolean hasFeature(Tuple<Integer, Integer> featureUnderObservation) {
    if (featureUnderObservation.getA() != 0) {
        if (getOpinionOn(featureUnderObservation.getA()).equals(featureUnderObservation.getB())) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

public boolean isEmpty() {
    boolean retVal = getKnownFeatures().isEmpty();
    return retVal;
}

public int getNumberOfFeatureChanges(ProductVector oldPV) {
    int retVal = 0;
    for (Integer feature : oldPV.getKnownFeatures()) {
        if (this.getOpinionOn(feature) != oldPV.getOpinionOn(feature)) {
            retVal++;
        }
    }
    return retVal;
}

public int getNumberOfNewFeatures(ProductVector oldPV) {
    ArrayList<Integer> featureList = this.getKnownFeatures();
    featureList.removeAll(oldPV.getKnownFeatures());
    return featureList.size();
}

PVFactory.java
package jEffCauSocialNetworkSimulator;

import java.util.ArrayList;
import bibliothek.util.container.Tuple;

public class PVFactory {
    private ArrayList<ProductVector> PVs = new ArrayList<ProductVector>();

    public void createPredefinedPVs(int NumberOfPVs) {
        PVs.clear();
        for (int i = 0; i < NumberOfPVs; i++) {
            ProductVector PV = new ProductVector(SimulationParameters.NumberOfFeatures, SimulationParameters.NumberOfVariants, SimulationParameters.NumberOfKnownFeaturesEffectuators, true);
            PVs.add(PV);
        }
    }
}
public Tuple<ProductVector, Integer> getRandomPredefinedPV()
{
    if (PVs != null)
    {
        int i = RandomHelper.getGenerator().nextInt(PVs.size());
        Tuple<ProductVector, Integer> retVal = new Tuple<ProductVector, Integer>();
        retVal.setA(new ProductVector(PVs.get(i)));
        retVal.setB(i + 1);
        return retVal;
    } else {
        return null;
    }
}

RandomHelper.java
package jEffCauSocialNetworkSimulator;

import java.util.ArrayList;
import java.util.Random;

public class RandomHelper {
    private static Random generator = null;

    public static Random getGenerator() {
        //checked
        if (RandomHelper.generator == null){
            RandomHelper.generator = new Random(SimulationParameters.RandomSeed);
        }
        return generator;
    }

    public static Integer drawfromProbabilityMassFunction(int[] histogram){
        //checked
        //Calculate sum as sum of all entries in ProbabilityMassFunction
        int totalSum = 0;
        for (int i : histogram){
            totalSum += i;
        }

        int totalSum != 0{
            int draw = getGenerator().nextInt(totalSum)+1;
            int runningSum = 0;
            int returnValue = 0;
            for (int i = 0; i < histogram.length; i++){
                runningSum += histogram[i];
                if(runningSum >= draw){
                    returnValue = i;
                    break;
                }
            }
            return returnValue;
        } else {
            return getGenerator().nextInt(histogram.length);
        }
    }

    public static void resetGenerator(){
        resetGenerator(SimulationParameters.RandomSeed);
    }

    public static void resetGenerator(long randomSeed){
        if(randomSeed>0){

    }

114
RandomHelper.generator = new Random(randomSeed);
} else {
    RandomHelper.generator = new Random();
}

public static int[] randomSequence(int range) {
    assert range > 0 : "Invalid range size";

    int[] destList = new int[range];
    ArrayList<Integer> sourceList = new ArrayList<Integer>();

    for (int i = 0; i < range; i++) {
        sourceList.add(i);
    }

    for (int i = 0; i < range; i++) {
        int sourcePosition = RandomHelper.getGenerator().nextInt(sourceList.size());
        destList[i] = sourceList.get(sourcePosition);
        sourceList.remove(sourcePosition);
    }

    return destList;
}

RandomNetworkGenerator.java
package jEffCauSocialNetworkSimulator;
import java.util.ArrayList;
import java.util.Random;
import repast.simphony.context.Context;
import repast.simphony.context.space.graph.NetworkGenerator;
import repast.simphony.space.graph.Network;

public class RandomNetworkGenerator extends EffectualNetworkGenerator implements NetworkGenerator<Object> {
    /**
     * @uml.property name="EdgeProbability"
     */
    protected float edgeProbability = 0.0F;

    public RandomNetworkGenerator(Context<Object> context, float EdgeProbability) {
        super(context);
        setEdgeProbability(EdgeProbability);
    }

    @Override
    public Network<Object> createNetwork(Network<Object> network) {
        this.network = network;
        return network;
    }

    @Override
    public Network<Object> createIntraAgentNetwork(Network<Object> network, Class<?> targetClass) {

```java
    this.network = network;
    return createInterAgentNetwork(network, targetClass, targetClass);
}

@Override
public Network<Object> createInterAgentNetwork(Network<Object> network, Class<?> originClass, Class<?> targetClass) {
    this.network = network;
    Random generator = RandomHelper.getGenerator();
    for (Object i : context.getObjects(originClass)) {
        if (i.getClass().getName().equals(originClass.getName())) {
            for (Object j : context.getObjects(targetClass)) {
                if (j.getClass().getName().equals(targetClass.getName())) {
                    if (!i.equals(j) && network.getEdge(i, j) == null && network.getEdge(i, j) == null) {
                        if (generator.nextFloat() <= getEdgeProbability()) {
                            NetworkEdge Edge = (NetworkEdge) network.addEdge(i, j);
                            Edge.setWeight(1);
                        }
                    }
                }
            }
        }
    }
    return network;
}

/**
 * Getter of the property <tt>EdgeProbability</tt>
 * @return Returns the edgeProbability.
 * @uml.property  name="EdgeProbability"
 */
public float getEdgeProbability() {
    return edgeProbability;
}

/**
 * Setter of the property <tt>EdgeProbability</tt>
 * @param edgeProbability The edgeProbability to set.
 * @uml.property  name="EdgeProbability"
 */
public void setEdgeProbability(float edgeProbability) {
    this.edgeProbability = edgeProbability;
}

@Override
public void step() {
    if (SimulationParameters.doCreateRandomLinks) { //checked, think about it contentwise!!!
        assert this.network != null : "Network not initialized yet!";
        float originalEdgeProbability = getEdgeProbability();
        setEdgeProbability(SimulationParameters.RNG_EdgeProbability * SimulationParameters.RNG_EdgeProbability);
        //Create Eff-Eff and Eff-EndCustomerLinks with very low probability
        createIntraAgentNetwork(network, Effectuator.class);
        createInterAgentNetwork(network, Effectuator.class, EndCustomer.class);
        setEdgeProbability(originalEdgeProbability);
    }
}
```
@Override
public Network<Object> addCreatorsWithFixedDegrees(Network<Object> network,
        ArrayList<Effectuator> creators, int creatorDegree) {
    assert 1<0 : "Not implemented!!!";
    return network;
}
}

SimulationParameters.java
package jEffCauSocialNetworkSimulator;

import bibliothek.util.container.Tuple;
import repast.simphony.engine.environment.RunEnvironment;

/**
   * @author Jan Willem Jansen
   */
public class SimulationParameters {
    public static repast.simphony.parameter.Parameters params =
RunEnvironment.getInstance().getParameters();

    public static long RandomSeed = 0;
    public static long RandomSeed_Network = 0;
    public static int NumberOfFeatures = 7;
    public static int NumberOfVariants = 2;
    public static int NumberOfKnownFeaturesEndcustomers = 4;
    public static int NumberOfKnownFeaturesEffectuators = 3;
    public static int NumberOfEffectuators = 10;
    public static int NumberOfCausators = 1;
    public static int NumberOfEndCustomers = 20;
    public static int NumberOfMarketResearchParticipants = (int)
Math.ceil(((float)NumberOfEndCustomers / 10.0F));
    public static float DocilityEffectuators = 0.2f;
    public static float PropagationProbability = 0.5f;
    public static float DistributionProbability = 0.0f;

    //Creator controls
    public static float PropagationProbabilityCreator = 0.5f;
    public static float DocilityCreator = 0.2f;
    public static int useCreators = 0;
    public static float goodnessOfIdea = 0.5f;
    //public static boolean doCentralizeCreator = false;

    //Initiator controls
    public static int useInitiators = 0;

    //network generator controls
    public static String NetworkGeneratorType = "BarabasiAlbert";
    public static float RNG_EdgeProbability = 1.0f;
    public static int BAG_EdgesPerStep = 1;
    public static boolean doCreatorDegreeSweep = false;
    public static int CreatorDegree = BAG_EdgesPerStep;

    //Behaviour controls
    public static boolean doMarketResearch = false;
    public static boolean doPropagateOwnidea = true;
    public static boolean doPickupNewFeatureFromOtherIdea = true;
    public static boolean doEvaluatePropagatedIdeas = true;
    public static boolean doPropagateOtherIdeas = true;
    public static boolean doCreateRandomLinks = false;
}
public static boolean doTransmitCreatorStrategyToProject = false;
public static boolean isActive_std = true;

//Simulation controls
public static boolean doEndIfSimIsStatic = true;
public static int AcceptableIdleTicks = 150;
public static int MinNumberOfSimSteps = 50;
public static int MaxNumberOfSimSteps = -1;

//Setup controls
public static String PVGeneratorType = "-";
public static int PVPathDependency_NumberOfTypes = 0;

//Measurement
public static Tuple<Integer, Integer> FeatureUnderObservation = new Tuple<Integer, Integer>(7, 1);
public static int VariantUnderObservation = 1;
public static float fixedEdgeWeight = 1.0f;
public static int NumberOfAnalysisThreads = 8;
public static boolean Network_analysis_required = true;
public static float Network_analysis_required_until = 10000f;
public static boolean CentralityAnalysisRequired = true;
public static boolean GroupCentralityAnalysisRequired = false;
public static boolean StructuralHolesAnalysisRequired = false;

/**
 * Load Parameters from simulation environment
 */
public static void LoadParameters() {
    //checked
    RandomSeed = (Integer) params.getValue("randomSeed");
    RandomSeed_Network = (Integer) params.getValue("RandomSeed_Network");
    NumberOfFeatures = (Integer) params.getValue("NumberOfFeatures");
    NumberOfVariants = (Integer) params.getValue("NumberOfVariants");
    NumberOfKnownFeaturesEndcustomers = (Integer) params.getValue("NumberOfKnownFeaturesEndcustomers");
    NumberOfKnownFeaturesEffectuators = (Integer) params.getValue("NumberOfKnownFeaturesEffectuators");
    NumberOfEffectuators = (Integer) params.getValue("NumberOfEffectuators");
    NumberOfCausators = (Integer) params.getValue("NumberOfCausators");
    NumberOfEndCustomers = (Integer) params.getValue("NumberOfEndCustomers");
    DocilityEffectuators = (Float) params.getValue("DocilityEffectuators");
    PropagationProbability = (Float) params.getValue("PropagationProbability");
    //DistributionProbability = (Float) params.getValue("DistributionProbability");
    NetworkGeneratorType = (String) params.getValue("NetworkGeneratorType");
    RNG_EdgeProbability = (Float) params.getValue("RNG_EdgeProbability");
    BAG_EdgesPerStep = (Integer) params.getValue("BAG_EdgesPerStep");
    NumberOfMarketResearchParticipants = (Integer) params.getValue("NumberOfMarketResearchParticipants");
    doCreatorDegreeSweep = (Boolean) params.getValue("doCreatorDegreeSweep");
    CreatorDegree = (Integer) params.getValue("CreatorDegree");

    //Creator controls
    PropagationProbabilityCreator = (Float) params.getValue("PropagationProbabilityCreator");
    DocilityCreator = (Float) params.getValue("DocilityCreator");
useCreators = (Integer) params.getValue("useCreators");
goodnessOfIdea = (Float) params.getValue("goodnessOfIdea");
//doCentralizeCreator = (Boolean)
params.getValue("doCentralizeCreator");

//InitiatorControls
useInitiators = (Integer) params.getValue("useInitiators");

//Behaviour controls
doMarketResearch = (Boolean) params.getValue("doMarketResearch");
doPropagateOwnidea = (Boolean) params.getValue("doPropagateOwnidea");
doPickupNewFeatureFromOtherIdea = (Boolean)
params.getValue("doPickupNewFeatureFromOtherIdea");
doEvaluatePropagatedIdeas = (Boolean)
params.getValue("doEvaluatePropagatedIdeas");
doPropagateOtherIdeas = (Boolean)
params.getValue("doPropagateOtherIdeas");
doCreateRandomLinks = (Boolean) params.getValue("doCreateRandomLinks");
doTransmitCreatorStrategyToProject = (Boolean)
params.getValue("doTransmitCreatorStrategyToProject");
isActive_std = (Boolean) params.getValue("isActive_std");

//Simulation controls
doEndIfSimIsStatic = (Boolean) params.getValue("doEndIfSimIsStatic");
AcceptableIdleTicks = (Integer) params.getValue("AcceptableIdleTicks");
MinNumberOfSimSteps = (Integer) params.getValue("MinNumberOfSimSteps");
MaxNumberOfSimSteps = (Integer) params.getValue("MaxNumberOfSimSteps");

//Setup controls
PVGeneratorType = (String) params.getValue("PVGeneratorType");
PVPathDependency_NumberOfTypes = (Integer) params.getValue("PVPathDependency_NumberOfTypes");

//Measurement
VariantUnderObservation = (Integer) params.getValue("VariantUnderObservation");
if (VariantUnderObservation>=0)
    FeatureUnderObservation = new Tuple<Integer, Integer>(NumberOfFeatures, VariantUnderObservation);
else
    FeatureUnderObservation = new Tuple<Integer, Integer>(0, VariantUnderObservation);

//fixedEdgeWeight = (Float) params.getValue("fixedEdgeWeight");
//NumberOfAnalysisThreads = (Integer) params.getValue("NumberOfAnalysisThreads");
Network_analysis_required = (Boolean)
params.getValue("Network_analysis_required");
Network_analysis_required_until = (Float) params.getValue("Network_analysis_required_until");
CentralityAnalysisRequired = true; // (Boolean)
params.getValue("Network_analysis_required");
GroupCentralityAnalysisRequired = true; // (Boolean)
params.getValue("GroupCentralityAnalysisRequired");
StructuralHolesAnalysisRequired = true; // (Boolean)
params.getValue("StructuralHolesAnalysisRequired");
package jEffCauSocialNetworkSimulator;

import java.util.concurrent.ExecutorService;
import repast.simphony.context.Context;
import repast.simphony.context.space.graph.ContextJungNetwork;
import repast.simphony.space.graph.Network;
import repast.simphony.space.graph.RepastEdge;
import edu.uci.ics.jung.algorithms.metrics.StructuralHoles;
import edu.uci.ics.jung.graph.Graph;

public class StructuralHolesAnalyses {
    private static Context<Object> context = null;
    private static Network<Object> network = null;
    private static StructuralHoles<Object, RepastEdge<Object>> structuralHoles = null;

    public static void getStructuralHolesMeasures(Context<Object> context,
                                                Network<Object> network, ExecutorService threadPool) {
        if (SimulationParameters.Network_analysis_required) {
            ContextJungNetwork<Object> N = (ContextJungNetwork<Object>) network;
            Graph<Object, RepastEdge<Object>> G = N.getGraph();
            structuralHoles = new StructuralHoles<Object, RepastEdge<Object>>(G,
                                  new EdgeWeightTransformer());
            for (Object i : context.getObjects(Effectuator.class)) {
                if (!i.getClass().getName().equals(Effectuator.class.getName())) {
                    continue;
                }
                EFFThread thread = new StructuralHolesAnalyses.EFFThread(i);
                threadPool.submit(thread);
            }
        }
    }

    public static class EFFThread implements Runnable {
        private Effectuator EFF = null;

        public EFFThread(Object i) {
            EFF = (Effectuator)i;
        }

        public void run() {
            float constraint = (float) structuralHoles.constraint(EFF);
            float effectiveSize = (float) structuralHoles.effectiveSize(EFF);
            float efficiencyStructHoles = (float) structuralHoles.efficiency(EFF);
            float hierarchy = (float) structuralHoles.hierarchy(EFF);
            EFF.setConstraint(constraint);
            EFF.setEffectiveSize(effectiveSize);
            EFF.setEfficiencyStructHoles(efficiencyStructHoles);
            EFF.setHierarchy(hierarchy);
        }
    }
}

package jEffCauSocialNetworkSimulator;

SystemBuilder.java
import java.util.ArrayList;
import bibliothek.util.container.Tuple;
import edu.uci.ics.jung.algorithms.shortestpath.DistanceStatistics;
import edu.uci.ics.jung.graph.Graph;
import repast.simphony.context.Context;
import repast.simphony.context.DefaultContext;
import repast.simphony.context.space.graph.ContextJungNetwork;
import repast.simphony.context.space.graph.NetworkBuilder;
import repast.simphony.dataLoader.ContextBuilder;
import repast.simphony.space.graph.EdgeCreator;
import repast.simphony.space.graph.Network;
import repast.simphony.space.graph.RepastEdge;

/**
 * @author Jan Willem Jansen
 */
public class SystemBuilder extends DefaultContext<Object> implements ContextBuilder<Object> {

    /**
     * @uml.property name="effectualNetworkGenerator"
     */
    private static EffectualNetworkGenerator effectualNetworkGenerator;

    /**
     * @uml.property name="network"
     */
    public static Network<Object> network;

    /**
     * @uml.property name="context"
     */
    public static Context<Object> context;

    //public int NumberOfIdleTicks = 0;

    //protected ArrayList<EndCustomer> endCustomers = new ArrayList<EndCustomer>();

    //protected ArrayList<Causator> causators = new ArrayList<Causator>();

    //protected ArrayList<Effectuator> effectuators = new ArrayList<Effectuator>();
public SystemBuilder() {
}

@Override
public Context<Object> build(Context<Object> context) {
    //checked
    if (context == null){
        context = new SystemBuilder();
    }
    context.clear();
    network = null;
    Effectuator.projectCounter = 0;
    String contextname = "EffectuationSimulation";
    context.setId(contextname);
    context.setTypeID(contextname);
    setContext(context);
    //context.add(this);
    //Load parameters from Runtime-Environment
    SimulationParameters.LoadParameters();
    RandomHelper.resetGenerator();

    //Initialise Networks
    initNetwork();

    //Last: Create Fitness Landscape
    FitnessLandscape myLandscape = new FitnessLandscape(
        SimulationParameters.NumberOfFeatures,
        SimulationParameters.NumberOfVariants);
    myLandscape.createRandomFitnesslandscape(0, 1);
    int leaveOutFeatures = 0;
    if (SimulationParameters.useCreators > 0){
        leaveOutFeatures = 1;
    }
    PVFactory myPVFactory = null;
    if (SimulationParameters.PVGeneratorType.equals("Predef")){
        myPVFactory = new PVFactory();
    }
    myPVFactory.createPredefinedPVs(
        SimulationParameters.PVPathDependency_NumberOfTypes);
    }
    //Create Agents
    //1st: Create End customers
    for (int i=1; i<= SimulationParameters.NumberOfEndCustomers; i++){
        EndCustomer ec = null;
        if (SimulationParameters.PVGeneratorType.equals("-")){
            ProductVector pv = new ProductVector(
                SimulationParameters.NumberOfFeatures - leaveOutFeatures,
                SimulationParameters.NumberOfVariants,
                SimulationParameters.NumberOfKnownFeaturesEndcustomers, true);
            pv.setFeature(SimulationParameters.FeatureUnderObservation.getA(),
                0);
            ec = new EndCustomer(context, network,"EC_"+Integer.toString(i), pv);
        }else{
            Tuple<ProductVector, Integer> tup =
                myPVFactory.getRandomPredefinedPV();
            ec = new EndCustomer(context, network,"EC_"+Integer.toString(i),
                new ProductVector(
                    SimulationParameters.NumberOfFeatures - leaveOutFeatures,
                    SimulationParameters.NumberOfVariants,
                    SimulationParameters.NumberOfKnownFeaturesEndcustomers, true,
                    new ProductVector(
                        SimulationParameters.NumberOfFeatures - leaveOutFeatures,
                        SimulationParameters.NumberOfVariants,
                        SimulationParameters.NumberOfKnownFeaturesEndcustomers, true))};
            ec = new EndCustomer(context, network,"EC_"+Integer.toString(i), pv);
        }
    }
ec = new EndCustomer(context, network, "EC_" + Integer.toString(i),
tup.getA());
    ec.setInitialPVType(tup.getB());
    context.add(ec);
    //endCustomers.add(ec);
}

// 2nd: Create Effectuators
for (int i = 1; i <= SimulationParameters.NumberOfEffectuators -
    SimulationParameters.useCreators - SimulationParameters.useInitiators;
    i++){
    Effectuator EFF = null;
    ProductVector pv = new
    ProductVector(SimulationParameters.NumberOfFeatures - leaveOutFeatures,
    SimulationParameters.NumberOfVariants,
    SimulationParameters.NumberOfKnownFeaturesEffectuators, false);
    pv.setFeature(SimulationParameters.FeatureUnderObservation.getA(),
    0);
    EFF = new Effectuator(context, network, "EFF_" + Integer.toString(i),
    pv,
    SimulationParameters.DocilityEffectuators,
    SimulationParameters.PropagationProbability,
    SimulationParameters.DistributionProbability,
    SimulationParameters.isActive_std);
    else{
    Tuple<ProductVector, Integer> tup =
    myPVFactory.getRandomPredefinedPV();
    EFF = new Effectuator(context, network, "EFF_" + Integer.toString(i),
    tup.getA(),
    SimulationParameters.DocilityEffectuators,
    SimulationParameters.PropagationProbability,
    SimulationParameters.DistributionProbability,
    SimulationParameters.isActive_std);
    EFF.setInitialPVType(tup.getB());
    context.add(EFF);
    //effectuators.add(EFF);
}

//2nd.A. Create "Creators" - The effectuators with a special feature
ArrayList<Effectuator> creators = new ArrayList<Effectuator>();
for (int i = 1; i <= SimulationParameters.useCreators; i++){
    ProductVector pv = new ProductVector();
    pv.setFeature(SimulationParameters.FeatureUnderObservation.getA(),
    SimulationParameters.FeatureUnderObservation.getB());
    Effectuator EFF = new Effectuator(context, network, "CRE_" + Integer.toString(i),
    pv,
    SimulationParameters.DocilityEffectuators,
    SimulationParameters.PropagationProbability,
    SimulationParameters.DistributionProbability, true);
    EFF.checkHasAdoptedThroughPickup();
    EFF.setCreator(true);
    EFF.setDocility(SimulationParameters.DocilityCreator);
    EFF.setPropagation_probability(SimulationParameters.PropagationProbability
    Creator);
    creators.add(EFF);
//Now add creators to SEPARATE LIST !!!!
//Probably repeated multiple times (but who cares :) )
if (SimulationParameters.goodnessOfIdea>=0)
{
    FitnessLandscape.clearFeature(SimulationParameters.FeatureUnderObservation.getA());
    FitnessLandscape.setFeatureVariant(SimulationParameters.FeatureUnderObservation.getA(),
    SimulationParameters.FeatureUnderObservation.getB(),
    SimulationParameters.goodnessOfIdea);
}

//2nd.B Create "Initiators" - the initially lone active effectuator accessing a "passive" network
for (int i=1; i<= SimulationParameters.useInitiators; i++)
{
    ProductVector pv = new ProductVector(SimulationParameters.NumberOfFeatures,
    SimulationParameters.NumberOfVariants,
    SimulationParameters.NumberOfKnownFeaturesEffectuators, false);
    Effectuator EFF = new Effectuator(context, network,
    "INT_"+Integer.toString(i), pv,
    SimulationParameters.DocilityEffectuators,
    SimulationParameters.PropagationProbability,
    SimulationParameters.DistributionProbability, true);

    //Add initiators separately to make use of "DegreeSweep"-Mechanism
    creators.add(EFF);
}

// 3rd: Create Causators (if any)
for (int i=0; i< SimulationParameters.NumberOfCausators; i++)
{
    Causator CAU = new Causator(context, network,
    "CAU_"+Integer.toString(i));
    context.add(CAU);
    //causators.add(CAU);
}

//Create Networks
// 0th: Initialise NetworkGenerator and load network seed
RandomHelper.resetGenerator(SimulationParameters.RandomSeed_Network);
if(!SimulationParameters.doCreatorDegreeSweep)
{
    context.addAll(creators);
}

if(SimulationParameters.NetworkGeneratorType.equals("BarabasiAlbert")){
    effectiveNetworkGenerator = new BarabasiAlbertNetworkGenerator(context);
} else if(SimulationParameters.NetworkGeneratorType.equals("Random")){
    effectiveNetworkGenerator = new RandomNetworkGenerator(context,
    SimulationParameters.RNG_EdgeProbability);
} else{
    throw new Error("No valid generator Type chosen");
}
//context.add(effectiveNetworkGenerator);

// 1st: Create Effectuator-Effectuator network + Effectuator-Customer network
boolean isConnected = false;
int i=1;
while(!isConnected){
    network = effectualNetworkGenerator.createIntraAgentNetwork(network, Effectuator.class);
    network = effectualNetworkGenerator.createInterAgentNetwork(network, Effectuator.class, EndCustomer.class);
    ContextJungNetwork<Object> N = (ContextJungNetwork<Object>)network;
    Graph<Object, RepastEdge<Object>> G = N.getGraph();
    double diameter = DistanceStatistics.diameter(G);
    if(diameter < Double.POSITIVE_INFINITY){
        isConnected = true;
    }else{
        RandomHelper.resetGenerator(SimulationParameters.RandomSeed+i);
        i++;
    }
}

//2nd: Add creators in a special way if necessary
if(SimulationParameters.doCreatorDegreeSweep){
    network =
    effectualNetworkGenerator.addCreatorsWithFixedDegrees(network, creators, SimulationParameters.CreatorDegree);
    context.addAll(creators);
}

//3rd: reset randomGenerator
RandomHelper.resetGenerator(SimulationParameters.RandomSeed);

AdvancedAnalyses myAdvancedAnalysis = new AdvancedAnalyses(context, network);
context.add(myAdvancedAnalysis);

System.out.println("Random Seed:" + String.valueOf(SimulationParameters.RandomSeed));
return context;
}

protected void initNetwork() {
    EdgeCreator<NetworkEdge, Object> edgeCreator = new
    EdgeCreator<NetworkEdge, Object>{
        public Class<NetworkEdge> getEdgeType() {
            return NetworkEdge.class;
        }
        @Override
        public NetworkEdge createEdge(Object source, Object target, boolean isDirected, double weight) {
            return new NetworkEdge(source, target, true, 1);
        }
    };
    //Initialise complete network
    NetworkBuilder<Object> netBuilder = new
    NetworkBuilder<Object>("CompleteNetwork", context, false);
    netBuilder.setEdgeCreator(edgeCreator);
    network = netBuilder.buildNetwork();
}

/**
* Getter of the property `<tt>network</tt>`
* @return Returns the network.
* @uml.property name="network"
*/

dublic Network `<tt>Object</tt>` getNetwork() {
    return SystemBuilder.network;
}

/**
* Setter of the property `<tt>network</tt>`
* @param network The network to set.
* @uml.property name="network"
*/
dublic void setNetwork(Network `<tt>Object</tt>` network) {
    SystemBuilder.network = network;
}

/**
* Getter of the property `<tt>context</tt>`
* @return Returns the context.
* @uml.property name="context"
*/
dublic Context `<tt>Object</tt>` getContext() {
    return SystemBuilder.context;
}

/**
* Setter of the property `<tt>context</tt>`
* @param context The context to set.
* @uml.property name="context"
*/
dublic void setContext(Context `<tt>Object</tt>` context) {
    SystemBuilder.context = context;
}
Part B. Research papers

Paper I – Individual vs. Collective Control in Effectual Social Networking: A Simulation Study

Paper II – Entrepreneurial Mingling Secrets: Investigating the Performance Impact of Network Structure for Control-Based Entrepreneurship using Agent-based Simulation

Individual vs. Collective Control in Effectual Social Networking: A Simulation

Study

Abstract

Who is really in control in control-based entrepreneurship? We investigate how and why the inter-subjective interaction behavior of effectual entrepreneurs affects emerging markets. Using agent-based computer simulation, we study the individual vs. collective impact of interaction behavior on emerging markets in an effectual network. Our findings contrast with the image of entrepreneurs single-handedly creating new markets and reveal the severely limited impact of individuals on emerging markets. Successful entrepreneurs require a collective of partners who act both docile and persistent in inter-subjective interactions. Moreover, we identify the significant impact of effectual transformation processes on the shape of emerging markets by disseminating new ideas beyond “project borders.”

Keywords: effectuation process, docility, agent-based computer simulation, effectual social networking
1 Introduction

The use of social networks, which is paramount for successful entrepreneurship, has been the subject of substantial research efforts (Hoang & Antoncic, 2003; Jack, 2010). Networks enable information exchange and foster resource acquisition (Hite, 2005; Slotte-Kock & Coviello, 2010). In this context, resources are both tangible and intangible: capital, emotional support, employees, and market knowledge (Hoang & Antoncic, 2003). Although detailed process models for the beneficial use of networks have yet to be determined, initial research suggests a prediction-based approach (Slotte-Kock & Coviello, 2010) that follows the idea of exploration and exploitation (Holland, 1975). In this process, entrepreneurs use social networks as a means to acquire all the resources required to pursue fixed, predefined goals. However, in uncertain environments, the application of this approach poses severe limitations as the definition of a goal worth pursuing is simply impossible. For such environments, effectuation (Sarasvathy, 2001) offers an alternative control-based decision logic that relies heavily on the use of social networks.

Effectuation is an alternative approach used by expert serial entrepreneurs. The approach is designed for use in environments that feature Knightian uncertainty (Knight, 1921), Marchian goal ambiguity (March & Olsen, 1982) and isotropy. Effectuation proposes exercising control in the shaping of markets rather than using prediction to optimally position one’s venture in an emerging market. Consequently, effectuation follows four principles that contradict prediction-based entrepreneurship. First, effectuation focuses on effects that are achievable with available means rather than the definition of a goal and the acquisition of all the required resources. Second, effectuation proposes keeping the potential loss associated with a project affordable instead of maximizing the expected return. Third, effectuation actively exploits contingencies instead of mitigating them. Lastly, effectuation proposes jointly co-creating markets with other market participants rather than viewing these participants as competitors.
Despite their contradictory networking approaches, case studies on both control- and prediction-based entrepreneurship place “the entrepreneur” at the center of the development of new markets (e.g., Dodgson, 2011). Moreover, initial research shows a positive performance impact of effectual social networking (Brettel, Mauer, Engelen, & Küpper, 2012; Read, Song, & Smit, 2009), which raises the question of the extent to which the process, the partners, and “the effectuator” can positively affect the emergence of markets.

Introduced in 2001 (Sarasvathy, 2001), effectuation has quickly drawn attention from the research community. In addition to scale development (Brettel et al., 2012; Chandler, DeTienne, McKelvie, & Mumford, 2011), effectuation was introduced to adjacent research (Goel & Karri, 2006; Sarasvathy & Dew, 2008) and tested quantitatively, which revealed a positive performance impact (Read, Song, et al., 2009; Wiltbank, Read, Dew, & Sarasvathy, 2009). In addition to theoretical broadening (Dew, Sarasvathy, Read, & Wiltbank, 2009; Read, Dew, & Sarasvathy, 2009; Sarasvathy, Dew, Read, & Wiltbank, 2008), effectuation was portrayed as a process (Dew, Read, Sarasvathy, & Wiltbank, 2008; Sarasvathy & Dew, 2005a), thus revealing the importance of inter-subjective interaction. Moreover, transformation mechanisms were recognized as an integral component of the effectuation process (Dew, Read, Sarasvathy, & Wiltbank, 2010).

As the discussion stabilizes on the relevant building blocks of effectuation and their positive performance impacts, research has focused on the mechanisms at work: effectuation is an agent-centric process that uses inter-subjective interaction and affects macro-level constructs such as market development. Consequently, questions have emerged regarding how micro-level interactions affect macro-level outcomes (Sarasvathy & Venkataraman, 2011). This consideration includes both the process and the actors. Although transformation types have been observed to be an important part of effectuation, the extent of and reasons for their positive impact remain unclear. Moreover, the behavioral impact of actors, that is, “the entrepreneur” and “the partners,” requires further research attention. What type of behavior does the effectuation process foster? How and why does certain interaction behavior affect performance? How and
why must we separate the behavior of “the entrepreneur” from that of “the partners”? To what extent can these entities exercise control, i.e., affect the shape of emerging markets?

Investigating these questions will help to improve the understanding of effectual entrepreneurship in multiple ways. First, this investigation deepens the understanding of effectuation and “under what circumstances [it] provide[s] particular advantages and disadvantages” (Sarasvathy, 2001, p. 249). Second, it focuses on “inter-subjective interaction[, which] is the very essence of the effectual process” (Dew et al., 2008, p. 50) and is regarded as a key research area of effectuation and entrepreneurship in general (Sarasvathy & Venkataraman, 2011). Third, although a significant number of studies have delivered helpful insights regarding the “content […], governance and structure” (Hoang & Antoncic, 2003, p. 1) of entrepreneurial social networks, this paper provides a much-needed “process-oriented study[ing]” (Hoang & Antoncic, 2003, p. 167). Lastly, process-based entrepreneurship research efforts are essential to transform entrepreneurship into a science of the artificial (Venkataraman, Sarasvathy, Dew, & Forster, 2012) – a method that can be tailored to specific entrepreneurial environments.

This study uses “heterogeneous agent-based computational modeling” (McKelvey, 2004), which is a method that is underrepresented in the social sciences but commonly used in engineering and the natural sciences. Computer simulation, as a “third way of doing science” (Axelrod, 2003, p. 1), enables the analysis of complex, intertwined, and non-linear processes (Davis, Eisenhardt, & Bingham, 2007). By enabling the collection of data at arbitrary points in time on all the modeled levels of detail, computer simulation fosters data analysis beyond empirical feasibility (Lévesque, 2004; McKelvey, 2004). Effectual social networking is optimally suited for computer-simulation research; given the “nascent/intermediate state” (Perry, Chandler, & Markova, 2012), the multi-actor setup and the intertwined, non-linear effectuation process, computer simulation is well suited to “contribute novel theory” (Davis et al., 2007, p. 482).

Following the roadmap of Davis et al. (2007, p. 482), we create a formal model of the effectuation process with all the involved entities and implement the model using the Repast
Simphony Framework (North, Howe, Collier, & Vos, 2007) to conduct three series of experiments in a Monte Carlo fashion (Davis, Eisenhardt, & Bingham, 2009). Using the example of shaping an emerging market by introducing a novel idea, we set up a typical entrepreneurial environment with effectual agents who interact with each other and the environment – represented by “end customer agents” – to shape an emerging market by fostering the adoption of a specific idea. We track the dissemination of this idea within the market and list reasons for its non-dissemination. By varying input parameters such as interaction behavior, we determine the extent of and reasons for performance variations. Although the contributions are theoretical, they may facilitate and simplify subsequent empirical research as researchers will know what to look for.

Our study aims to offer the following four contributions and explanations: a formal model of effectuation, the importance of transformation mechanisms for effectuation, the significant impact of the collective interaction behavior of partners, and the limited impact of the individual interaction behavior of the focal entrepreneur. To the best of our knowledge, this model is among the first models of effectual decision making.

Our first contribution is a formal model of effectuation that unifies available principles (Sarasvathy, 2001), dimensions (Chandler et al., 2011), process descriptions (Dew et al., 2008; Sarasvathy & Dew, 2005a), and transformation mechanisms (Dew et al., 2010). This stylized model requires precise terminology and will therefore help to mitigate ambiguous processes and definitions. An examination of the current debate on whether entrepreneurial opportunities are “made” vs. “created” or “found” vs. “discovered” (Alvarez & Barney, 2013; Shane, 2012; Venkataraman et al., 2012) emphasizes this need. Moreover, we hope to encourage the use of computer simulation in effectuation and entrepreneurship research in general, as promoted by leading researchers (Davis et al., 2007; Harrison, Carroll, & Carley, 2007; Lévesque, 2004; McKelvey, 2004).
Our second contribution suggests the importance of transformation mechanisms for effectuation. Our findings reveal that although market saturation doubles using transformation, this mechanism is used rather rarely. Transformation has a trigger effect and is paramount for controlling the emergence of a market. This concept is counter-intuitive from a prediction-based entrepreneurship perspective because it includes giving away new ideas to competitors. However, from a control-based perspective, providing valuable ideas to other market participants facilitates the shaping of an emerging market, and the immediate loss through strengthening the competition can be mitigated in the long run through a larger number of adopters. An illustrative historical example of this scenario is the introduction of potatoes to Prussia, Germany around 1750: when farmers refused the cultivation of a plant with poisonous leaves, Friedrich II founded guarded plantations and instructed soldiers to look the other way when curious farmers tried to steal the roots for cultivation.

Our third contribution suggests the strong impact of collective interaction behavior. Behaving docilely in negotiations and being persistent in approaching new potential partners both have a strong positive impact on the dissemination of a new idea in an emerging market. An examination of the reasons for non-dissemination indicates that docile behavior not only fosters successful negotiations among effectual entrepreneurs but also improves the inclusion of the conscious and sub-conscious demands of end customers, which thus improves the general utility of an opportunity. In summary, our contribution provides reasons why “[e]ffectuation assumes docility as a fundamental behavioral construct applicable to all partners” (Dew et al., 2008, p. 49). Moreover, our findings reveal that persistence in approaching new partners is an import quality in actual and potential partners. Examining the reasons for non-dissemination reveals that partners have a crucial multiplication function in the shaping of emerging markets, which contrasts with their role as passive entities in existing process descriptions (Dew et al., 2008; Sarasvathy & Dew, 2005a).
Our last contribution indicates the surprisingly small impact of “the entrepreneur.” Our findings reveal that individual persistence in approaching potential partners has a positive but weak impact in comparison to collective persistence. Moreover, docile behavior has no positive impact on shaping emerging markets. “This contrasts with the image of the persistent entrepreneur who holds on against all odds and against all skepticism to bring an idea to fruition” (Wood & McKinley, 2010, p. 71).

2 A review of the effectuation process

Prior to the creation of a formal model of effectuation, we review the available literature and extract the relevant process information by focusing on inter-subjective interaction and possible parameters. Information on the effectuation process can be grouped into four categories. First, the description of effectuation began with “principles” (Sarasvathy, 2001) and “views” (Sarasvathy & Dew, 2005b, p. 390) that were later used as a basis for scale development (Brettel et al., 2012; Chandler et al., 2011) and performance evaluation (Dew, Read, Sarasvathy, & Wiltbank, 2009; Wiltbank et al., 2009), as well as to organize further research (Perry et al., 2012). Second, researchers linked the actions of effectuators and the aforementioned principles in order to create a process-based view of effectuation (Dew et al., 2008; Read, Dew, et al., 2009; Sarasvathy & Dew, 2005a). Third, transformation mechanisms were identified as an integral effectual technique (Dew et al., 2010). Lastly, to fill the gaps in the literature on effectuation processes that emphasize network interaction, we consulted reviews on entrepreneurial networking (Hoang & Antoncic, 2003; Jack, 2010).

Based on these studies, we created an integrated process model, as illustrated in figure 1. Similar to prior process descriptions (Dew et al., 2008, p. 49), we chose an ego-centric description that an effectual entrepreneur can follow in a step-by-step fashion.
Step 1: Means

In addition to “Who I am” and “What I know,” social networks (“Whom I know”) are a key resource for entrepreneurs using effectuation logic. Effectuation teaches us to begin with available means, which refers to the assessment of individual preferences, networks and resources. Effectuation proposes to focus on effects that can be achieved with available means (Sarasvathy, 2001).

Step 2: What can we, as a project, do?

Based on the idea of pre-commitments, entrepreneurs who use effectuation logic can join projects to co-create an opportunity (Sarasvathy & Dew, 2005a). Unlike prior process descriptions, we emphasize joint action by including the creation of a joint opportunity on a team, as described in the literature (Dew et al., 2008).

Step 3: Interaction with known contacts

To create or finalize an opportunity, entrepreneurs who use effectuation logic create a “project” (Dew et al., 2008, p. 50), approach known contacts, and present their current
opportunity to negotiate pre-commitments. The effectuation literature usually treats this action as a mere prelude to subsequent steps (Dew et al., 2008). However, “the ask”\(^6\) is an integral step in the effectuation process and holds a previously unmentioned degree of freedom for effectual interaction behavior: the decision to (not) approach a potential partner. Based on personality traits (“Who I am”) and the affordable loss of time and effort, the persistence with which effectual entrepreneurs approach potential partners is part of their individual interaction behavior. Although this persistence might be related to the “proactiveness” dimension of entrepreneurial orientation (Lumpkin & Dess, 1996, p. 146), this quality is less focused on leadership. Whereas the effectuation literature focuses on the use of network contacts for direct interaction, the network literature mentions another important function: the exchange and relay of information. Therefore, we include the “relay of information” as a second important process activity that is consistent with the idea of the contingency principle (Sarasvathy, 2001).

Except for a willingness to exchange pre-commitments to avoid situations in which “non-customers drive the decision process” (Dew & Sarasvathy, 2007), the effectuation literature does not limit the type of customers who can be approached. This lack of restrictions enables the inclusion of non-entrepreneurs, i.e., end customers who intend to use rather than actively develop an opportunity.

**Step 4: Negotiation with potential partners**

To further the finalization of an opportunity, effectual entrepreneurs engage in negotiations regarding pre-commitments. These negotiations can be initiated by themselves or by direct and indirect network contacts. Acting on the “contingency principle” (Sarasvathy, 2001) effectual entrepreneurs do not dismiss other projects with limited similarities, but actively exploit these unforeseen opportunities as a source of unconventional ideas. Because of uncertainty, effectual entrepreneurs dismiss the idea of market research and focus on the negotiation of a series of pre-

\(^6\) As referred to by Sarasvathy at the “Effectuation Research and Teaching Conference 2013,” Lyon.
commitments. Here, the negotiations will primarily focus “on the characteristics of the project […] rather than the type and quantity of returns” (Dew et al., 2008, p. 50) because the market itself is emerging with the opportunity.

Although the literature on entrepreneurial and effectual negotiation is sparse (Sarasvathy & Venkataraman, 2011), docility has been identified as “a fundamental behavioral construct” (Dew et al., 2008, p. 50). Introduced by Simon (1990), docility is defined as “the tendency to depend on suggestions, recommendations, persuasion, and information obtained through social channels as a major basis for choice“ (Simon, 1993a, p. 156). Simon later added that “[w]e are highly susceptible to social influence and persuasion, susceptibility that I will call docility. I use the term ‘docility’ here in its sense of teachability or educatability -- not in its alterative sense of passivity or meekness” (Simon, 1997). Consequently, we recognize docility to be another important behavioral degree of freedom for effectual entrepreneurs.

**Step 5: Transformation**

The negotiation of pre-commitments is not the only way to improve an opportunity through effectuation. Based on a case study, Dew et al. (Dew et al., 2010) list 9 types of transformations that expert serial entrepreneurs apply to their opportunities to shape emerging markets in their favor. Contrary to negotiations, this action does not lead to a contribution of new contacts or resources. To create a formal model of effectuation, we focus on those types of interactions (“deletion and supplementation” and “composition and decomposition”) that use external stimuli provided through the network to transform an opportunity. These types of interactions do not require any form of “genius,” which is neither a prerequisite for the application of effectuation nor a concept that can be properly formalized.

**Step 6: Concretion of opportunity**

Over time, the increased number of partners creates “two concurrent dynamic cycles” (Sarasvathy & Dew, 2005a, p. 1), which leads to an increase in committed resources and the finalization of the joint opportunity.
Step 7: New contacts

New partners will likely lead to new contacts for the project both internally (because new partners are likely to be acquainted with current partners) and externally (because new partners may know potential partners beyond the reach of the former group of partners). Because the process is iterative, new partners will approach their contacts in subsequent rounds of the effectuation process until the “effectual transformation [is] complete” (Sarasvathy & Dew, 2005a, p. 549).

3 A formal model of effectual social networking

Modeling effectual social networking requires a formal representation of all the relevant actors, objects, and processes of effectual networking. This representation includes entrepreneurs who use effectuation logic, their opportunities, and the environment. Entrepreneurs who use effectuation logic form a social network that allows them to approach each other. According to the literature (Dew & Sarasvathy, 2007), effectual entrepreneurs cooperate only with partners who are willing to make pre-commitments. Therefore we represent the environment as a number of end customer agents that are part of the social network and can be approached by effectual entrepreneurs. Moreover, we pay particular attention to the implementation of uncertainty, which is a major prerequisite for effectuation (Sarasvathy, 2001).

3.1 General setup of the model

Our formal model consists of effectual entrepreneurs who aim to shape an emerging market in their favor. Starting with a loosely defined individual opportunity, effectual entrepreneurs approach their known contacts to negotiate pre-commitments and to create a joint opportunity in a joint project. The entrepreneurs execute the effectuation process in each time step in a turn-based fashion. They finalize their opportunities by including the ideas of new project partners and shape a new market using a control-based approach that involves effectual entrepreneurs and end customers who represent the environment. The simulation ends when it reaches a stable
state, i.e., no new contacts are created, and no changes occur in any opportunity within 10 consecutive time steps.

In the subsequent chapters, we first introduce all the objects and actors in our formal model and then introduce the actions that they perform.

3.2 Objects and actors

3.2.1 Artifact

The artifact is our stylized version of an opportunity that the entrepreneur finalizes using effectuation logic (Sarasvathy, 2001), representing the individual means (“Who I am” and “What I know”), i.e., the desires and interests of each effectual entrepreneur. Following the product description of arbitrary merchandise, e.g., a life vest, we formalize an artifact as a list of (feature: variant)-tuples. For example, a life vest could be described as follows: \{(color: red), (weight: 500 g, (fabric: polyester), (air capacity: 6 liters))\}. Mathematically speaking, we formalize an artifact \(A\) as a set of (feature, variant)-tuples of variable length, as presented in equation 1. Each feature can appear only once in an artifact, and the number of variants per feature and the total number of features are limited, discrete, and strictly positive. Notably, these maximum numbers are not known by the agents. Hence, Knightian uncertainty is introduced into the model because agents know neither the number nor the distribution of existing features and variants.

\[
A = \{(f_a, v_a), (f_b, v_b), \ldots \} \quad \forall i \ f_i \in \{1..f_{\text{max}}\}, \forall i \ v_i \in \{1..v_{\text{max}}\}, \forall a, b \in \{1..f_{\text{max}}\} \ f_a \neq f_b
\]

\[
f_{\text{max}} \in \mathbb{N}^+, \ v_{\text{max}} \in \mathbb{N}^+
\]

In equation 2, we define \(v(f, A)\) as the variant of feature \(f\) in artifact \(A\), if available. The variant “0” is reserved for “unknown feature.” In our example, the variant of “color” of our life vest is “red,” and the variant of the feature “blinking lights” is “0” because it was not yet defined.

\[
v(f, A) = \begin{cases} 
0, & \exists f \ (f, v) \in A \\
v, & \exists f \ (f, v) \in A
\end{cases}
\]
The set $F(A)$ in equation 3 represents the set of all known features in $A$. In the life-vest example, this would be the set \{color, weight, fabric, air capacity\}.

$$F(A) = \{ f_i \mid \forall v_i \ (f_i, v_i) \in A \} \quad (3)$$

One of these (feature: variant)-combinations plays a special role in our simulation: because we aim to investigate the shaping of emerging markets, we study the propagation of one arbitrarily chosen combination, which represents our new idea under study. The ability to shape emerging markets can then be measured as the level of dissemination of this idea.

### 3.2.2 Effectual agent

We model entrepreneurs who use effectuation logic as agents pursuing the task of artifact finalization. Every effectual agent has an artifact, which is partially pre-populated with randomly drawn (feature: variant)-combinations during the initialization of the simulation. The effectual process is executed in a turn-based simulation including interactions with other effectual agents and “end customer agents” who represent the environment. Although the extant literature implies that effectual entrepreneurs can “declare the effectual transformation complete and begin competing in alternative markets” (Sarasvathy & Dew, 2005a, p. 549), we focus on a single round of market development and end the simulation once the development of a market is completed. Moreover, we assume that each partner contributes sufficient resources to drive transformations and negotiations. Therefore, we omit an explicit modeling of resources and potential constraints and assume that every action an effectual agent takes fulfills the affordable loss principle.

One effectual agent plays a special role in our formal model. This agent is called the “inventor” and provides a starting point for the new idea under study. Although this agent behaves like any other effectual agent, the agent’s artifact consists of only the new idea under study, a single (feature: variant)-combination, which is exclusively received during initialization.
To investigate the effect that individuals can have on the spread of new ideas, we allow for an individual adjustment of the inventor’s level of docility and persistence in the experiments.

3.2.3 End customer agent

End customer agents represent the market demand in our current model. Formalized as agents in our model, these agents fulfill the important role of providing insights into what customers want and regard as useful. The effectuation literature often mentions the important role of end customers in examples such as the customer-investors of U-Haul, who helped to both shape and grow a one-man, one-truck company into a company that held “essentially 100 percent market share in the newly created do-it-yourself moving industry” (Sarasvathy, 2001, p. 248). More recent examples include the users of crowd-funding platforms such as Kickstarter (see Belleflamme, Lambert, & Schwienbacher, 2013). These customers represent the market demand to the entrepreneurs who ask for funding. In part, this role is also assumed by entrepreneurs who use effectuation logic and their prior knowledge (“What I know”).

Therefore, we formalize end customers as “passive” agents with two distinct properties: these customers can participate in the negotiation of pre-commitments when engaged by effectual agents. End customer agents then represent the market demand, including both conscious and sub-conscious demands in these negotiations. Like effectual agents, all end customer agents have an artifact that represents their conscious demands. To represent sub-conscious demands, we use the concept of market utility (see next chapter). This concept enables an end customer agent to compare two opportunities and identify the preferred one based on market utility. Based on Henry Ford’s famous quote regarding faster horses, i.e., conscious demands, we assume that upon presentation, an end customer will choose a car over faster horses because a car has greater utility even if it contradicts the conscious demand.

During initialization, the artifact of each end customer agent is partially pre-populated with (feature: variant)-combinations. While the features are drawn randomly, the respective variant is drawn to probabilistically represent the market demand. For example, if the artifact represents a
life vest, the utility of a yellow vest is much higher than the utility of a black one. Therefore, the yellow variant is much more likely than the black variant to be assigned to the color feature of an artifact.

3.2.4 Market utility

We formalized the concept of market utility as an “NK-fitness landscape,” which is a well-known approach in management research that employs simulation studies (Ganco & Agarwal, 2009; Gavetti & Levinthal, 2000; Levinthal, 1997). Market utility assigns a “utility value” to all possible artifacts that represents the utility of these artifacts if presented to an end customer. In accordance with effectuation theory, we implemented the simplest version of an NK-fitness landscape, which regards products as “near decomposable” (Sarasvathy, 2003). The total utility of an artifact A in our fitness landscape is defined in equation (4) as the sum of the utility of each feature-variant combination.

\[ U(A) = \sum_{f_i \in F(A)} u((f_i, v_i)) + \sum_{f_i \notin F(A)} u((f_i, 0)) \]

\[ \forall f_i \in \{1..f_{\text{max}}\}, v_i \in \{0..v_{\text{max}}\} \] \[ u((f_i, v_i)) \in [0,1] \] randomly

We randomly initialize our utility function \( u((f_i, v_i)) \) for all possible feature-variant-combinations. Moreover, we assign a value to features that are not available in the artifact. For the new idea under study, we define a fixed utility value of 1.0 if a feature occurs in the artifact and 0.0 if it does not occur. The utility of an artifact that represents the aforementioned life vest would, for example, be calculated by awarding points to known features, e.g., “color,” with high scores for “yellow” and low scores for “black.” We also assign points for missing features, e.g., “entertainment system,” which might be useless in emergency situations but could be considered an additional luxury for snorkeling trips. Therefore, the artifact scores points because of the absence of certain features. All points are then summed to calculate the total utility \( U(A) \) of an artifact.
3.2.5 Social network

We initialize the network among effectual agents and between effectual and end customer agents as an undirected scale-free network according to the generation procedures introduced by Barabási and Albert (Barabasi & Albert, 1999). We assume undirected networks because each of the two acquainted parties is able to approach the other one. Numerous studies have shown that scale-free networks best capture the structure of real-world networks (Aiello, Chung, & Lu, 2001; Barabasi & Albert, 1999; Broder et al., 2000; Jeong, Mason, Barabási, & Oltvai, 2001; Newman, 2000). Scale-free networks assume that the likelihood $p$ of having $k$ known contacts assumes a negative exponential shape ($p \sim k^{-\alpha}$).

Using the algorithm of Barabási and Albert, we first create a network between effectual agents. Therefore, the algorithm cycles through all effectual agents and creates exactly one tie originating from this effectual agent $i$. The target agent $j$ of the tie is probabilistically chosen as indicated in equation 5:

$$p(j) \sim \beta + \text{deg} \text{ree} (j)^{\alpha}, \alpha = 1, \beta = 0.1, i \neq j$$

The function $\text{degree}(j)$ refers to the number of contacts known by the agent $j$. Second, all the end customer agents are attached to the previously created network in the same way. Connections between end customer agents are omitted because they would serve no purpose for these passive agents.

3.3 Simulation mechanics

3.3.1 Behaviors of effectual agents

An effectual agent can exhibit the following four behaviors: approaching a known contact to present the current state of its artifact (Step 3a), relaying information on the artifacts of other effectual agents to known contacts (Step 3b), negotiating strategic alliances by exchanging pre-commitments (Step 4), and inheriting the ideas of other effectual agents’ artifacts through transformation (Step 5). All of these behaviors are influenced by the effectual agents’ levels of
docility and persistence. Both constructs are formalized as a variable that represents low levels of persistence and docility as 0.0 and high levels of persistence and docility as 1.0.

3.3.2 Approaching known contacts (Step 3a)

Effectual agents can approach known contacts to present the current state of their artifact. We formalized this behavior as transferring copies of the artifact. For each contact, the effectual agent executes this behavior probabilistically using the persistence parameter as an execution probability. The effectual agent can approach all known contacts except for its partners once per turn. This behavior is executed at the beginning of the simulation and every time the agent’s artifact changes. The recipient processes the proposed artifact in the next time step. In this way, isotropy is induced because the receiving agents cannot determine a general direction or trend coming from the artifacts of other effectual agents.

3.3.3 Relaying received artifacts to known contacts (Step 3b)

Effectual agents can relay information regarding other agents’ artifacts. We formalized this behavior as probabilistically transferring unaltered copies of incoming artifacts to all known contacts except partners. This behavior is triggered by every incoming artifact and is probabilistically executed using the persistence parameter as an execution probability. Again, the recipient processes the relayed artifact in the next time step. To avoid multiple receptions of the same artifact, a list of previous recipients is transferred along with the artifact. In this way, the repeated acceptance of the same artifact can be avoided.

3.3.4 Negotiation of pre-commitments (Step 4)

“Even the literature that is directly focused on negotiations has mostly neglected new venture creation processes” (Sarasvathy & Venkataraman, 2011, p. 126). Despite the absence of a concrete algorithm, the literature (Sarasvathy & Dew, 2005a) proposes multiple requirements for such a formalization as follows: (a) allow the negotiation between two or more parties, (b) ensure that ideas supported by a majority of participants are more likely to prevail, (c) support an individual acceptance or refusal of negotiation results to ensure both self-selection into a project
and the consideration of pre-commitments, and (d) ensure that negotiations are triggered by the approach of an effectual agent, as described in the “effectual cycle” (Dew et al., 2008).

As illustrated in figure 2, we formalize the negotiation process as a simple, three-stage probabilistic negotiation. First, the algorithm determines all the relevant partners, i.e., the project partners of the approaching and the approached agent. Subsequently, the algorithm gathers information on the artifacts of all the involved agents. Second, the algorithm creates a combined list of occurring features and scans the list to create histograms of the frequency of the partners’ preferred variants. Third, the algorithm creates a proposed negotiation result that includes all the known features. The variant of each feature is determined probabilistically using the histograms as a non-normalized discrete probability mass function. For example, if 1 partner prefers the “color”-feature to be “red” and 9 prefer it to be “blue,” the artifact will be blue with a likelihood of 90%. Lastly, the algorithm proposes the negotiation result to each participant to receive feedback. A negotiation result is accepted only if all the participants agree on its adoption. In the event of an adoption, the algorithm replaces the artifact of each participant with the negotiation result and creates new network ties among all the new project partners.

After a negotiation result is created, each effectual agent will individually compare the result to the state of its own artifact and decide whether to adopt the proposal. Because “[e]ffectuation assumes docility as a fundamental behavioral construct” (Dew et al., 2008, p. 498), we propose a docility-based formalization of the acceptance process that follows a simple rule: the less similar the negotiation result is to the effectual agent’s own artifact, the more docility is required to adopt it.
We formalized this rule as the calculation of the share of “conflicting” features in relation to the total number of features that the artifact of the deciding effectual agent holds. A conflicting feature is a different variant of the same property, e.g., a “red vest” instead of a “blue vest.” We assume the following: the larger the share of conflicts over known features for an entrepreneur who uses effectuation logic, the more docile an entrepreneur who uses effectuation logic must be to accept these changes. Therefore, in our formal representation, an effectual agent accepts a negotiation result if the relative share of conflicting features is lower than its docility. If an effectual entrepreneur $i$ with artifact $A$ examines the negotiation result $R$, the entrepreneur can accept this result if equation (6) is true.

$$\frac{\left| \{ v(f_i, A) \neq v(f_i, R) \mid f_i \in F(A) \} \right|}{|F(A)|} \leq \text{docility}_i$$

(6)

3.3.5 Transformation of artifact (Step 5)

As presented in the literature review, we formalize artifact transformation as a single abstract behavior as follows: the process is triggered by a previously unsuccessful negotiation with the owner of the proposed artifact. Initially, the algorithm compiles a list of features that are exclusive to the proposed artifact. Second, it randomly chooses one of these features. Third, the algorithm includes the chosen feature-variant combination probabilistically into the receiving
effectual agent’s artifact using the receiving agent’s docility as an execution probability. Lastly, the receiving agent initiates a negotiation for adoption with its partners.

This formalization implies the assumption that a certain amount of docility is required to transform an artifact based on external information. Moreover, we assume that the adoption of a previously unknown feature is significantly more likely than the change of a known feature, which likely represents both the agent’s and its partners’ preferences.

3.3.6 End customer agents’ assessment of market utility

During the negotiation, end customer agents can access the fitness landscape that represents the market utility exactly once. When confronted with the decision to adopt a negotiation proposal, end customer agents can compare the utility of the proposal and their current artifact. This restricted access formalizes the notion of goal ambiguity, i.e. that customers “are not quite sure of their own […] preferences” (Sarasvathy & Dew, 2005b, p. 401). The access restriction is essential to maintain the state of uncertainty in the effectual environment. Knightian uncertainty is maintained because the number of design alternatives and their respective success probabilities cannot be determined by the end customer agent. Goal ambiguity is maintained because end customer agents are unaware of their sub-conscious desires. Isotropy is maintained because the conscious desires that the end customer agent enters into a negotiation algorithm do not necessarily correspond to its sub-conscious demands.

4 Simulation experiments

We designed three simulation experiments to investigate the importance of transformation, the impact of collective interaction behavior, and the impact of individual behavior.

If not stated otherwise, we use the following set of standard parameters: we created a population of 60 end customer agents and 30 effectual agents. Except for the inventor, all the agents received an artifact with 5 randomly selected features that were drawn from 9 available features with 2 respective variants. The docility of all the effectual agents was set to 0.2, and the
persistence was set to 0.6. Each simulation was repeated 1,000 times per parameter setting, with randomly initiated artifacts and randomly generated networks.

4.1 The importance of effectual transformation

In our first experiment, we investigated the impact of transformation on shaping emerging markets.

As reflected in figure 3a, we measured the level of adoption of the idea under study over time allowing and disallowing transformation. The results show that the shape of the adoption curve follows a “logistic function” (Verhulst, 1845) also known as an s-curve. The s-curve is a well-known function to describe diffusion processes (Mahajan & Peterson, 1985) and has been used as an input in other computer simulations of innovation and social networks (Abrahamson & Rosenkopf, 1997). Simulation studies on distribution mechanisms such as critical mass (Marwell, Oliver, & Prahl, 1988; Oliver, Marwell, & Teixeira, 1985) or threshold models (Granovetter, 1978; Macy, 1991) have indicated similar outcomes.

More importantly, we observe a significant increase in adoption levels when comparing scenarios that allow and disallow transformation. While a scenario without effectual transformation reaches an average adoption level of approximately 30 agents, this figure increases to approximately 60 agents when transformation is allowed.

Additionally, figure 3b shows the number of adoptions of the idea under study over time by reason – negotiation or transformation. While enabling transformation doubles the adoption level, figure 3b shows that only approximately 5 adoptions occur directly due to transformation behavior.
4.2 The impact of collective interaction behavior

In our second simulation experiment, we analyzed the impact of collective inter-subjective interaction behavior on shaping emerging markets. Therefore, we conducted a single parameter variation of collective docility (0.01 and 0.99) and persistence (0.01 and 0.99). Although these extreme settings are unlikely to occur in reality, we used these settings to exemplify the effects that could occur. In addition to the effect of these variations on the adoption level, we investigated the rate of occurrence of the following four possible reasons for the containment of new ideas: (a) a breakdown in negotiations because of the limited docility of one or more effectual agents, (b) a breakdown in negotiations because of a negative utility impact for one or more end customer agents, (c) a failure to propagate the state of an artifact including the new idea, and (d) a failure to adopt the new idea under study via transformation from a presented artifact after unsuccessful negotiations. Figure 4 displays the number of events that lead to the containment of the new idea under study over time for our standard set of parameters. The rate of failed negotiations and non-adoptions begins at a low rate and increases exponentially, reaching a
peak after 5-6 time steps and then decreasing dramatically as the simulation approaches a stable state near time step 21.

---

Insert figure 4 here

---

Table 1 shows both the average level of adoption of the idea under study and the average total occurrence of reasons for non-dissemination for all 4 parameter variation experiments and for the standard set of parameters. The results are presented separately by parameter in the two subsequent chapters.
## 4.2.1.1 Table 5: Impact of effectual behavior on the containment of a new idea

<table>
<thead>
<tr>
<th>Measure</th>
<th>Experiment</th>
<th>Standard parameters</th>
<th>High docility</th>
<th>Low docility</th>
<th>High persistence</th>
<th>Low persistence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. number of adopters</td>
<td></td>
<td>59.2</td>
<td>78.5</td>
<td>13.6</td>
<td>76.4</td>
<td>2.8</td>
</tr>
<tr>
<td>Reason for containment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Docility</td>
<td></td>
<td>24.2</td>
<td>0.4</td>
<td>31.3</td>
<td>31.0</td>
<td>0.5</td>
</tr>
<tr>
<td>(b) Utility</td>
<td></td>
<td>17.4</td>
<td>6.3</td>
<td>20.9</td>
<td>23.9</td>
<td>1.1</td>
</tr>
<tr>
<td>(c) Not propagated</td>
<td></td>
<td>27.0</td>
<td>17.0</td>
<td>6.7</td>
<td>2.9</td>
<td>8.4</td>
</tr>
<tr>
<td>(d) Not adopted</td>
<td></td>
<td>16.4</td>
<td>0.0</td>
<td>18.6</td>
<td>19.3</td>
<td>0.7</td>
</tr>
</tbody>
</table>

## 4.2.2 The impact of collective docility

High collective docility increases the average dissemination of the idea under study significantly from 59.2 to 78.5. Consequently, non-dissemination based on docility (a) and non-adoption (d) deteriorates nearly completely. Moreover, non-dissemination based on non-propagation (c) and insufficient utility (b) decreases.

Low collective docility leads to opposing effects: in addition to a starkly lower average adoption rate of 13.6, it increases the rates of containment events due to limited docility (a) and the non-adoption via transformation (d). Additionally, low collective docility leads to an increased number of containment events because of limited utility (b). However, low collective docility reduces the number of non-propagation events (c).

## 4.2.3 The impact of collective persistence

The level of collective persistence has a significant effect on the adoption level, varying it from 2.8 to 76.4 for low/high settings. The number of non-dissemination events due to docility (a), utility (b), and non-adoption (c) increases with high collective persistence and decreases with
low collective persistence. The number of containment events due to non-propagation (c) decreases with high collective persistence, as expected.

4.3 The impact of individual interaction behavior

In our third experiment, we investigated the impact of individual inter-subjective interaction behavior. Keeping the general levels of docility and persistence constant, we investigated the impact of varying levels of interaction parameters of the inventor of the idea under study. Although it is difficult to change the levels of collective docility and persistence, entrepreneurs who use effectuation logic can consciously modify their own levels of docility and persistence to affect the adoption level of their new idea. In our experiment, we simulated this scenario through a variation of the inventor agent’s docility and persistence parameters while keeping all the other parameters in accord with our standard model. In this experiment, we focused on the impact on the adoption level because the impact on the reasons for containment did not reveal additional insights beyond the aforementioned results. The results reflected in figures 5a and 5b show the number of agents who adopted the new idea under study at the end of the simulation under variations of the inventor agent’s docility (fig. 5a) and persistence (fig. 5b) parameters.

```
---
Insert figure 5 a+b here
---
```
As indicated by figure 5a, changing the level of the inventor agent’s docility has no positive impact on the adoption level of the idea under study. Although a low level of docility decreases the level of adoption, an increase above the collective level of docility has nearly no effect on the level of adoption.

As reflected in figure 5b, the level of individual persistence has a small positive effect on the level of adoption of the idea under study. However, with an increase from ~60 (standard persistence) to ~63 (maximum persistence), the effect is significantly smaller than the effect of the aforementioned collective persistence. As expected, low individual persistence decreases the adoption to ~55.

5 Propositions for the effectuation process, individual and collective interaction behavior

In this chapter, we present and support five propositions deduced from the results of the simulation experiments presented in chapter 4. We discuss the impact of the effectuation process and the impact of individual and collective interaction behavior on the shaping of emerging markets.

5.1 The impact of effectual transformations

Proposition 1: Effectual transformation mechanisms convert entrepreneurs outside the project into multipliers for a new idea. This behavior propagates ideas for other projects and thus significantly increases the number of adopters.

In our simulation experiments, allowing the application of transformation nearly doubles the level of adoption of the idea under study. Surprisingly, our results show that transformation rarely is the reason for propagation. Therefore, how does a mechanism that is rarely used double the rate of success?

The explanation is counter-intuitive from a control-based perspective: by allowing transformation and thus foregoing direct control regarding the use of his or her idea, an effectual
entrepreneur fosters the shaping of an emerging market by relying on the creativity and strength of others. By allowing transformation, other effectual entrepreneurs become the users and active promoters of a new idea and use negotiation to promote the idea as part of their opportunities, which is indicated by the nearly doubled rates of propagation via negotiation. The surprising success of this method not only underlines its vital importance for effectual entrepreneurship but also contradicts the current “best practice” of prediction-based entrepreneurs. Prediction-based entrepreneurship usually exercises a tight regime of control over new ideas. The restrictive use of patents and secrecy through non-disclosure agreements (Dushnitsky & Shaver, 2009) is a vital part of practices to create and maintain a lead over competing firms. The transformation approach can be seen as an instantiation of Simon’s “intelligent altruism” (Simon, 1993b). In addition to being a conclusive theoretical proposition, transformation also exhibits successful practical uses: when IBM released its personal computer (PC) in 1981, the company also provided the “IBM PC Technical Reference Manual” that included circuit schematics, BIOS source codes and all other information required to create replicas. By releasing this information, IBM enabled the competition to quickly create its own designs (“IBM clones”) based on IBM’s key ideas, e.g., x86 processor commands or BIOS. By enabling other partners, IBM played a crucial role in the rapid growth of the PC market and managed to benefit despite the loss of control.

5.2 The impact of collective interaction behavior

In addition to the effectuation process, the results reveal a significant impact of collective interaction behavior.

*Proposition 2: Increased collective docility results in more cooperative negotiation behavior and stronger end customer involvement. Consequently, increasing levels of collective docility lead to an increase in the adoption of new ideas.*
The simulation results clearly indicate the positive impact of high collective docility on shaping emerging markets in favor of a new idea. Moreover, the results show decreased levels of non-dissemination because of insufficient docility and utility. These findings reveal the reasons behind the positive impact of collective docility. It is clear that increased docility reduces non-dissemination because of limited docility. However, the interaction with utility is counterintuitive at first sight because effectual entrepreneurs do not optimize for “general” utility. Moreover, the effectual agents in our formal model have no means to access, assess, estimate, or process the utility level of an opportunity; only end customer agents can assess the utility of an opportunity. Consequently, the positive impact of increased collective docility can be motivated only by the interaction of effectual entrepreneurs and end customers. Increased docility leads to more openness toward both other entrepreneurs and end customers. Consequently, the conscious ideas of end customers are more likely to be included and accepted by entrepreneurs. Therefore, the utility of an opportunity for end customers increases, and the market demand is shifted toward the opportunity. Consequently, low docility decreases the success of shaping emerging markets and increases the number of unsuccessful negotiations for two reasons. First, negotiations break down because of the rejection of negotiation proposals as a direct consequence of low docility. Moreover, because entrepreneurs who use effectuation logic refuse to adopt the ideas of end customers, end customers are more often confronted with opportunities of inferior utility, which they reject.

These findings are highly relevant for entrepreneurship research for two reasons. First, although the benefits of a docile environment have been previously theorized (Simon, 1993a) as an abstract benefit for each individual, we contribute to this research through a determination of the relevant effectual processes and a detailed investigation of the positive impact of docility.

Proposition 3: Increased collective persistence improves the information availability regarding new ideas. As a result, increasing levels of collective persistence lead to an increase in the adoption of new ideas.
High collective persistence significantly increases the success of shaping emerging markets. However, this factor also increases the number of non-dissemination events because of insufficient docility and utility. Intuitively, a high number of containment events would result in a low level of adoption. The counter-intuitively high level of adoption is a result of the effectual networking mechanisms. High levels of persistence lead to a large number of opportunity presentations and relays of information about opportunities. Consequently, information travels farther in the network and more often leads to the initiation of negotiations. As reflected in the high number of containment events, this behavior increases the number of successful negotiations. The behavior also helps to shape an emerging market by making relevant information more accessible and fosters the discovery of like-minded cooperation partners.

Conversely, low collective persistence decreases adoption levels in two ways. First, the propagation of the idea itself is inhibited through low propagation. Moreover, re-propagation is also inhibited, which effectively decreases the size of the available social network.

These results are in line with findings on the importance of information flow in social networks (Burt, 1992; Granovetter, 1973). Moreover, the increased availability of information generates more situations in which entrepreneurs who use effectuation logic can embrace contingencies, as proposed by Sarasvathy (Sarasvathy, 2001). Therefore, high collective persistence can be seen as both a driver of information flow and a provider of contingency opportunities. Although the current literature treats the persistence to approach potential stakeholders as a given, these findings highlight the importance of this behavioral parameter and emphasize its importance beyond the individual. Although the individual benefits of persistence are obvious, these results show that an environment with persistent potential partners is vital to successfully shaping emerging markets for an individual entrepreneur who uses effectual logic.
5.3 The impact of individual interaction behavior

Our last simulation experiment focused on the impact of the docility and persistence of an individual on shaping an emerging market. In contrast to their collective counterparts, the impact of both parameters on the adoption level is limited.

*Proposition 4: Elevating individual docility beyond collective levels does not positively influence the outcomes of negotiations and hence has no positive impact on the adoption of new ideas.*

Contrary to collective docility, our findings reveal that the level of docility of an individual cannot be used to increase the adoption of a new idea. With the exception of low levels, the level has no impact on the level of adoption. Although this finding appears to be counter-intuitive to the presented literature and our previous findings, a thorough review of the mechanisms at work reveals the underlying reasons. Effectual negotiations are dominated by the least docile participant. The principle of accumulating pre-commitments leading to increased available resources and concretization of the opportunity (Dew et al., 2008) implies that negotiation results must be accepted unanimously. Therefore, the individual with the lowest docility will ultimately decide on the acceptance of a negotiation result. In turn, an individual can only influence the negotiation outcome negatively by choosing a very low docility and rejecting it.

These findings highlight the limited individual control in effectuation and an important feature of effectual negotiations. The effectuation literature and the entrepreneurship literature (Dew et al., 2008; Sarasvathy & Dew, 2005a, 2005b; Simon, 1993b) highlight the general importance of docility but fail to present a precise mechanism. Using our formal model of effectuation and the results regarding individual impact, we can shed light on the mechanics at work, differentiate the impact of individual and collective docility, and explain non-linear impacts, as in the case of low individual docility.
Proposition 5: Increased individual persistence improves the information availability regarding new ideas. Thus, increasing levels of individual persistence have an (incremental) positive effect on the adoption of new ideas.

With regard to individual persistence, our findings reveal a measurable but limited impact on shaping of an emerging market. As expected, being persistent in approaching one’s own contacts has a positive impact; however, the reach of an individual entrepreneur is limited, and therefore, the surplus in terms of information dissemination is small compared to the effect of increased collective persistence. Given the previous results, this finding is not surprising; however, the finding corrects the standpoint of the literature, which often portrays “the persistent entrepreneur who holds on against all odds and against all skepticism to bring an idea to fruition” (Wood & McKinley, 2010, p. 71).

5.4 Limitations and avenues for further research

Our study and its model hold limitations that provide avenues for further research in the following three areas: theory development, empirical confirmation, and model enhancement. Regarding the development of effectuation theory, we are confident that the presented formal model of effectuation will serve as a starting point for the further formalization of effectuation logic. As demonstrated in the literature review, vital parts of the effectuation process regarding effectual communication, negotiation, and the withdrawal of pre-commitments are insufficiently addressed by the current literature. Moreover, we propose broadening the effectuation research agenda. In addition to current research calls to focus on the entrepreneur as a person (Perry et al., 2012), we suggest that the “network of entrepreneurs who use effectuation logic” is an entity that must be well investigated to understand the complex dynamics at work. How do the networking positions of entrepreneurs who use effectuation logic or groups thereof affect the performance of a network? How do known network effects, e.g., weak links (Granovetter, 1973) or structural holes (Burt, 1992), affect a network of entrepreneurs who use effectuation logic? Which network
structures promote or discourage the efficient cooperation of entrepreneurs who use effectuation logic?

The empirical confirmation of the proposed effects is also of key importance. The presented propositions were derived solely from a theory-based computer simulation. Given the nascent state of effectuation theory, the propositions require careful validation through empirical research. Moreover, a significant amount of groundwork is required to create and validate the required constructs for previously unstudied parameters such as docility and persistence.

Lastly, our formal model provides multiple improvement opportunities. In particular, the inclusion of resources would facilitate a more comprehensive formalization of effectuation logic. However, developing a resource creation and consumption model will prove to be a complex task. Moreover, the determination of proper resource consumption values for each effectual and causal action will require extensive empirical research. In addition to the inclusion of resources, the introduction and inclusion of additional performance measures represent another avenue for further research. Measures regarding market demand, opportunity variance, or the network position of individuals and projects would foster further simulation-based studies of the network effects of effectuation and could help elucidate the underlying mechanics of effectuation.

6 Conclusion

The overwhelming impact of collective behavior and the limited impact of individual docility and persistence provide a contrast to the all-too-heroic perspective that the literature often adopts regarding the effectual entrepreneur who single-handedly creates markets through the accumulation of partners and resources (Sarasvathy, 2001). Instead, our study promotes the entrepreneur’s responsibility to choose a docile environment, attract persistent partners, and foster joint action through the use of social networks. Moreover, effectual transformation mechanisms enable other market participants to shape an emerging market in the entrepreneur’s favor. By forgoing control, this powerful tool enables control-based entrepreneurs to implant their ideas in parts of the market far beyond their reach.
REFERENCES


Entrepreneurial Mingling Secrets: Investigating the Performance Impact of Network Structure for Control-Based Entrepreneurship using Agent-based Simulation

“Knowing many people is good for business, keeping them apart is even better”, captures the essence of research regarding the position and shape of entrepreneurial social networks. Current research focuses on high-level relationships, lacking understanding of underlying processes. Consequently, its applicability to novel approaches such as control-based entrepreneurship, a decision logic designed for highly uncertain environments, is limited. Fostering co-creation and strong involvement of partners, control-based entrepreneurship is conceptually at odds with current networking strategies emphasising transactional relationships, arbitrage, and brokering between contacts. In this study, we therefore re-evaluate current theories regarding network position and shape for control-based entrepreneurship. We use a computer simulation of effectuation, a prototype of control-based entrepreneurship. We reveal the starkly different mechanics that lead to a similarly positive impact of network position yet completely contrasting results for network shape. Proposing “tertius iungens” as an alternative theoretical foundation, we demonstrate how control-based entrepreneurship reorganises social networks towards a dense web with few structural holes, high personal centrality and highly constrained stakeholders.

Keywords: effectuation, control-based entrepreneurship, social network, centrality, structural holes, tertius gaudens, tertius iungens, agent-based computer simulation

1. Introduction

The creation and utilisation of social networks are of key importance for entrepreneurs (Jack, 2010; Slotte-Kock & Coviello, 2010). As a means of organisation, social networks allow entrepreneurs to gain access to a wide variety of tangible and intangible resources provided by other parties (Hoang & Antoncic, 2003). A large body of research has determined the benefits of an appropriate network structure, effective governance mechanisms, and access to content through social networks. A central position (Brajkovich, 1994) and sufficiently large network (Aldrich & Reese, 1993) are structural characteristics that increase venture performance because they increase the accessibility of resources held by other parties. Researchers have argued for the
theoretical importance of both strong and weak relationships, or “ties”, (Granovetter, 1973) and have obtained largely supportive findings through empirical research (Aldrich, Rosen, & Woodward, 1987; Hite, 2005; Jack, 2005; Rowley, Behrens, & Krackhardt, 2000). A more general theory uses the occurrence of “structural holes” (Burt, 1992) to explain the beneficial impact of network structures by determining their efficiency, and effectiveness. With respect to governance, systems that rely on social mechanisms, such as trust (Larson & Starr, 1993) and reputation, have proven more effective than contractual systems (Jones, Hesterly, & Borgatti, 1997).

Despite these significant recent advances in research on entrepreneurial social networks, the understanding of the individual entrepreneurial networking process and its effects on the organisation of the network itself remains limited. The majority of research focuses on high-level input-output effects, leaving the underlying processes opaque. Available process descriptions typically focus on the network in general and discuss phases of network development in broad terms (Slotte-Kock & Coviello, 2010). In addition, recent studies have criticised the fact that the initial starting position was not considered (Witt, 2004) as well as the lack of research on network development over time (Jack, Dodd, & Anderson, 2008). Moreover, key theoretical contributions (Burt, 1992; Granovetter, 1973) assume an entrepreneurial behaviour that uses predictive planning, causal logic, and the classical exploration-exploitation scheme (Holland, 1975). Despite the widespread assumption of the universality of this prediction-based approach, a growing body of studies reveals that expert serial entrepreneurs prefer an alternative control-based entrepreneurship approach. In this study, we use effectuation (Sarasvathy, 2001) as an exemplary prototype.

Effectuation is a non-predictive logic that strives to control future market development by shaping it. With respect to network creation and utilisation, effectuation relies on co-creation and collaboration to jointly create a new market (Dew, Read, Sarasvathy, & Wiltbank, 2008;
Sarasvathy & Dew, 2005a). Initial studies indicate a positive effect of these practices – including the collaborative networking approach – on venture performance (Read, Song, & Smit, 2009). In light of these discoveries, in this study, we re-evaluate findings regarding entrepreneurial network structure and deepen the understanding of the interaction dynamics of entrepreneurs using control-based logic and their networks as well as their impact on venture performance. We examine network structure and venture performance using agent-based simulations based on three research questions: (1) How does the network position of an entrepreneur using control-based decision logics impact venture performance over time? (2) How does the network shape of an entrepreneur using control-based decision logics impact venture performance over time? (3) How do entrepreneurs using control-based decision logics re-organise their social networks over time?

2. Literature review: the performance impact of the social network

In this chapter, we provide an introduction to effectuation theory as a representative of control-based entrepreneurial decision logic. We subsequently present an in-depth review of the effectuation process literature to derive a formal model for the subsequent simulation experiments. We conclude this chapter with a presentation of current theories and empirical results regarding the performance impact of network structure, i.e., network position and shape. We expose inconsistencies with control-based entrepreneurship logic and propose possible explanations, methods of unification, and alternative theories.

2.1. Effectuation as an entrepreneurial expert decision logic

Effectuation was introduced as a decision logic as a result of case studies on new venture creation under Knightian uncertainty (Knight, 1921) with expert serial entrepreneurs (Sarasvathy, 2001). These experts approached the creation of new ventures in a manner fundamentally different from traditional approaches which employ prediction to best position a venture in an
emerging market.

Contrary to the traditional prediction-based approach called causation, effectuation is a control-based logic (Wiltbank & Dew, 2006) that aspires to shape future market developments rather than forecast them. Effectuation relies on four principles (Sarasvathy, 2001): first, effectuation focuses on the use of readily available means, contrasting the causation approach to first set a goal and subsequently acquire all required means. Second, effectuation proposes project selection based on maintaining an affordable potential loss rather than maximising the expected return. Third, effectuation proposes to embrace contingencies as an improvement opportunity rather than treating it as a deviation that requires mitigation. Finally, effectuation proposes viewing other market participants as potential partners rather than competitors. In fact, effectuators organise a network of partnerships through the negotiation of pre-commitments and the exchange of new ideas to drive the finalisation of an artefact (Sarasvathy & Dew, 2005a), typically a product, service, project, or venture.

Introduced by Sarasvathy (Sarasvathy, 2001), effectuation has developed into an important area of entrepreneurship research (Venkataraman, Sarasvathy, Dew, & Forster, 2012). Despite significant advances, effectuation is still considered to be in a nascent state (Perry, Chandler, & Markova, 2012). Originally derived from thought experiments, effectuation has expanded into a comprehensive decision logic including a process model (Sarasvathy & Dew, 2005a). Subsequent refinements (Dew et al., 2008; Dew, Read, Sarasvathy, & Wiltbank, 2010) led to the inclusion of effectual transformation mechanisms (Dew et al., 2010). Effectuation was introduced to adjacent research streams, for example trust (Goel & Karri, 2006; Karri & Goel, 2008; Sarasvathy & Dew, 2008a). A particularly fruitful debate regarding similarities and differences of Austrian economics and effectuation helped to delineate the concepts of non-predictive control and creativity (Chiles, Bluedorn, & Gupta, 2007; Chiles, Gupta, & Bluedorn, 2008; Sarasvathy & Dew, 2008b). Scale development has been initiated (Brettel, Mauer,
Engelen, & Küpper, 2012; Chandler, DeTienne, McKelvie, & Mumford, 2011), and quantitative studies revealed increased venture performance for business angels (Wiltbank, Read, Dew, & Sarasvathy, 2009) and new ventures in general (Read et al., 2009).

The principles of effectuation were derived from scenarios with high uncertainty. More precisely, Sarasvathy et al. (2008) identified three distinct preconditions: Knightian uncertainty, Marchian goal ambiguity, and isotropy. Although these conditions typically occur “naturally” in studies regarding new ventures, simulation studies such as this one must place particular emphasis on these preconditions to produce valid results.

2.2. Effectual use of social networks – a review of known processes and behaviours

Entrepreneurs using effectual logic focus on available means: “who I am”, “what I know”, and “whom I know” (Sarasvathy & Dew, 2005a). Consequently, social networks (“whom I know”) are essential in acquiring “new means” and “new goals” (Sarasvathy & Dew, 2005a). Although interaction with other market participants is regarded as “the very essence of the effectual process” (Dew et al., 2008: 50), research on inter-subjective interaction remains sparse (Sarasvathy & Venkataraman, 2011). The following review provides an overview of known effectual processes and behaviours. We created an integrated effectual process map based on available studies.

As illustrated in Figure 1, social networks are relevant to the effectuation process in four ways. First, effectuators approach contacts in their social network to negotiate pre-commitments (Sarasvathy, 2001). This typically implies the prior disclosure of the artefact on which an approaching effectuator is working. Second, effectuators use social networks as a mechanism to gather and relay information. Although this behaviour is not exclusive to effectuators, it exposes network participants to new ideas, which is an important principle of effectuation (Sarasvathy, 2001). Third, effectuators engage in the actual negotiation of pre-commitments and organisation of joint projects (Sarasvathy & Dew, 2005a). Finally, effectuators use transformation
mechanisms to develop the artefact on which they are working (Dew et al., 2010; Sarasvathy et al., 2008). Although not every transformation necessitates social networks, we focus on transformations that are initiated by external stimuli. In contrast to negotiations, transformations lead to concretion of the artefact without the acquisition of new means.

2.2.1. Presentation of artefacts

Effectuation relies on interactions with other market participants to create a market opportunity (Sarasvathy, 2001). Effectuation uses the negotiation of pre-commitments to mitigate the risk of mistaking an uninterested party for a potential customer (Dew & Sarasvathy, 2007). The negotiation of such pre-commitment is initiated by approaching a known contact (“whom I know”) and presenting the current state of the artefact (Dew et al., 2008). The literature treats this first step as a prelude to negotiations (Dew et al., 2008; Sarasvathy & Dew, 2005a) Assuming that effectual entrepreneurs will approach only a part of their known contacts, we argue to include the persistence with which effectual entrepreneurs approach known contacts in a formal model of effectuation. It is an individual decision or personality trait reflected in their means base (“who I am”) (Sarasvathy, 2001: 253).

2.2.2. Relay of information

The relay of information is a central aspect of social networks (Wasserman & Faust, 1994). Although it is not exclusive to entrepreneurs using effectual logic, the dissemination of information through networks plays a crucial role for them. The incidental reception of new information regarding other market participants is an important source of contingency that effectuation teaches us to embrace (Sarasvathy, 2001). Thus, the relay of such information is an important by-product of effectual networking. Although the existence and positive impact of information exchange have been studied extensively (Hoang & Antoncic, 2003), the exact
process steps used by entrepreneurs using effectual logic have yet to be determined (Sarasvathy & Venkataraman, 2011).

2.2.3. Negotiation of pre-commitments

The negotiation of pre-commitments is the key use of social networks for entrepreneurs employing effectual logic (Sarasvathy & Dew, 2005a). The results of a negotiation are two-fold: “new means” and “new goals” (Sarasvathy & Dew, 2005a: 543). “New means” refers to the surplus of both tangible and intangible resources that the other party committed to the development of the opportunity, whereas “new goals” refers to the further concretion of the artefact under development. As the market is still emerging, these negotiations focus “on the characteristics of the project […] rather than the type and quantity of returns” (Dew et al., 2008: 50). In this manner, the resource base and level of concreteness grow iteratively over time, a fact that is referred to as “two concurrent dynamic cycles” (Sarasvathy & Dew, 2005a: 1). Moreover, the exchange of pre-commitments is more than a one-time negotiation: participants organise a joint project and are part of it until they “declare the effectual transformation complete” (Sarasvathy & Dew, 2005a: 549). This high level of stakeholder involvement is conceptually different from the “classical” transactional use of social networks. Stakeholders are continuously involved in the development of a market opportunity rather than “just” providing resources in exchange for funds or equity.

“Effectuation assumes docility as a fundamental behavioral construct applicable to all stakeholders” (Dew et al., 2008: 50). Introduced by Simon (1990), docility describes “the tendency to depend on suggestions, recommendations, persuasion, and information obtained through social channels as a major basis for choice” (Simon, 1993: 156). Not to be interpreted as “passivity or meekness” (Simon, 1997), it is an important characteristic and must be included in a formal representation of effectual negotiation.
2.2.4. Transformation of artefacts

In (2008) Sarasvathy et al. presented alternative mechanisms that effectual entrepreneurs use to further develop artefacts without the negotiation of pre-commitments. Although these transformation mechanisms lead to “new goals” in the sense of artefact concretion, they do not contribute “new means” because no new stakeholders are gathered.

Based on their case studies, Dew et al. (2010) list nine different transformation types applied by entrepreneurs using effectual logic. Because the present study focuses on the effect of network position and shape, we focus on transformation types that involve the use of social networks. These types (e.g., “deletion and supplementation”, “composition and decomposition”, “free associating”) can be triggered by external stimuli provided by social networks. Moreover, we argue that the application of transformation mechanisms – if triggered by external stimuli – is subject to personal docility: similar to negotiations, effectuators change their artefact based on externally provided information. Then, a similar chain of considerations in which docility plays an important role is set in motion. The only notable difference is that no new stakeholders and resources are gathered in the process.

2.3. Performance impact of the position and shape of entrepreneurial social networks

Hoang and Antoncic (2003) list three research areas regarding the impact of entrepreneurial networks: content, governance, and structure. Focusing on structure as the best researched category, we will now review and discuss findings regarding two key qualities that characterise the social network of entrepreneurs: position (size, centrality) and shape (structural holes).

2.3.1. Performance impact of network position

The position within a network is usually defined by the measures of centrality or size. Researchers have identified a variety of benefits stemming from centrality and size, all of which have a common theme: the network enables access to resources outside the direct reach of the
entrepreneur. The size of an entrepreneur’s network increases the quantity (Adler & Kwon, 2002; Batjargal, 2003) and variety (Greve & Salaff, 2003; Greve, 1995) of accessible resources and improves the likelihood of their successful acquisition (Semrau & Werner, 2013). In this context, “resources” include both tangible resources, such as funding (Vanacker, Manigart, Meuleman, & Sels, 2011; Zimmer & Aldrich, 1987) and human resources (Freeman, 1999), and intangible resources, such as emotional support (Brüderl & Preisendörfer, 1998), information to recognise opportunities (Birley, 1985), and access to production capacity and distribution channels (Brown & Butler, 1995). Using all accessible resources, the depicted prediction-based entrepreneur will gather knowledge of customer preferences and exploit a new idea in such a way that the demands of a predefined target group are optimally satisfied.

Control-based entrepreneurship, however, proposes a different approach to networking. Controlling the shape of a market requires influencing market participants and tightly including them to foster joint action. Despite the difference in approach, the impact of network position in control-based entrepreneurship has not been discussed in the literature. Aside from the generally positive performance impact (Read et al., 2009), to our knowledge, control-based social networking has not been discussed at all. At best, we can assume that increased centrality, resulting in a larger means base, improves the performance of control-based entrepreneurship by enabling access to more potential partners. Thus, the stark differences of the control- and prediction-based approach warrant closer investigation of the performance impact of network position.

2.3.2. Performance impact of network shape

Hoang and Antoncic (2003) identify “bridging weak ties” and “structural holes“ as key theoretical constructs with regard to network shape. “Bridging weak ties” (Granovetter, 1973), refer to the finding that that acquaintances loosely related to an actor are more likely to provide non-redundant information. Granovetter’s conclusions have been extended by Burt (1992) and
subsumed under his “structural holes” paradigm. Not unlike the idea of an “electron hole” in solid state physics (Weller, 1967), a structural hole describes the absence of a tie between two actors. According to Burt (1992), entrepreneurs profit from a network rich in structural holes. Hoang and Antoncic (2003) list multiple studies (Baum, Calabrese, & Silverman, 2000; Krackhardt, 1995; Zaheer & McEvily, 1999) that empirically support Burt’s theory. Burt proposes that entrepreneurs should actively manage their social network to be as efficient (non-redundant) and effective (focusing on contacts with an abundance of structural holes) as possible. Burt proposes a strategy enabling entrepreneurs to actively benefit from an effective and efficient network called “tertius gaudens”. Originally introduced by Simmel (1896), tertius gaudens refers to a party who benefits from a conflict between two other parties. Burt argues that entrepreneurs should re-organise their network toward non-redundant contacts to engage in information arbitrage and benefit from competition for information. Because the exploited parties are not aware of each other, the entrepreneur is not constrained by the risk of disclosure through direct communication. However, the depicted entrepreneurs behave significantly differently from control-based expert serial entrepreneurs. Moreover, Burt’s strategy relies on prediction-based planning and assumes in an environment of relative certainty.

In (2005), Obstfeld proposed a behavioural model that opposed Burt’s tertius gaudens approach. Revisiting Simmel’s work on triads (1950) – the effects of constellations of three actors – Obstfeld proposed an approach that focuses on actively filling social holes to foster collaboration and the emergence of innovative projects. His approach ultimately reduces the efficiency of networks and increasingly constraints an entrepreneur by decreasing his or her degree of freedom in negotiations. Building on Burt’s findings that the tertius gaudens strategy fosters “good ideas” but ultimately negatively impacts their implementation (Burt, 2004), Obstfeld argues that the benefits of collaborative innovation and the decreased mobilisation efforts might outweigh the benefits of Burt’s tertius gaudens strategy.
Given the emphasis of both effectuation and the “tertius iungens” approach on collaboration, control-based entrepreneurship strategies, such as effectuation, appear conceptually closer to “tertius iungens” than “tertius gaudens” (Sarasvathy & Dew, 2005b). However, effectuation is classified as an approach suitable for radically innovative ideas (Sarasvathy, 2001) and could thus benefit from a network shaped in favour of a “tertius gaudens” entrepreneur. In effect, control-based strategies have not yet been associated with either strategy. Moreover, it is unclear how a network shaped in (dis)favour of a tertius gaudens entrepreneur affects control-based strategies.

3. Computer simulations as a method

This study uses agent-based computer simulations to examine the impact of network position and shape. As a “third way of doing science” (Axelrod, 1997), computer simulations can create unique insights, particularly for theories in a nascent state (Davis, Eisenhardt, & Bingham, 2007), such as the control-based effectuation logic (Perry et al., 2012). In addition to their flexibility, computer simulations allow for data collection at a level of detail and accuracy unparalleled by empirical research (McKelvey, 2004). This is an important aspect, particularly for social network studies, because “about half of what people report about their own interactions is incorrect in one way or another” (Wasserman & Faust, 1994: 57). Once implemented, computer simulations allow for “virtual experiments” (Davis et al., 2007) to study available theory under varying conditions with minor effort. This approach enables the development of a deeper understanding of theory, particularly because it can be tested under boundary conditions – areas typically associated with sparse data in empirical studies.

Although computer simulations are rarely used in entrepreneurship research, “several influential research efforts have used simulation as their primary method” (Harrison, Carroll, & Carley, 2007). March (1991) investigates the relation between exploration and exploitation in
organisational learning using a form of agent-based simulation. Ganco and Agarwal (2009) use computer simulations to study the performance impact of environmental turbulence and firm characteristics for various industries. Davis et al. (Davis, Eisenhardt, & Bingham, 2009), used computer simulation to investigate the interplay between firm structure and environmental complexity.

Using the step-based approach of Davis et al. (2007), we develop a formal model of effectuation emphasising the use of social networks based on expert interviews and the literature review presented in chapter 2. We implemented the model using the simulation framework Repast Simphony (North, Howe, Collier, & Vos, 2007) and the JUNG library (O’Madadhain, Fisher, Smyth, White, & Boey, 2005) to calculate network measures. This setup enables automatized data collection for individual agents at each time step.

4. A formal model of effectual social networking

Modelling effectual social networking requires a formal representation of all relevant actors, objects, and processes of effectual networking. This includes effectuators and their artefacts as well as end customers, representing the environment. Effectuators and end customers form a social network that allows effectuators to approach both end customers and other effectuators. According to the literature (Dew & Sarasvathy, 2007), effectuators cooperate only with stakeholders who are willing to make pre-commitments. We paid special attention to the implementation of uncertainty because this is a major prerequisite for effectuation (Sarasvathy et al., 2008).

In brief, the simulations are performed as follows: starting at time step one, all effectual agents begin presenting their artefacts to known contacts. The approached agents probabilistically relay

\[\text{(Partly taken from my paper “Individual vs. Collective Control in Effectual Social Networking”)}\]
the received artefact to known contacts. Moreover, they engage in negotiations with the approaching effectual agent. If an agreement is reached, both agents agree on a joint artefact and continue their search for additional stakeholders together. If no agreement is reached, the engaged party can decide to transform its artefact based on the information gained from the presented artefact. Over time, the artefacts are refined, and the simulation converges to a setting of multiple stable projects. The simulation is stopped after 40 steps, which is a sufficient amount of time to reach a stable state.

4.1. Objects and actors

4.1.1. Artefact

The term artefact refers to a product, service, project, or goal being finalised by an effectuator (Sarasvathy, 2001). Following the product description of arbitrary merchandise, e.g., a mobile phone, we formalised an artefact as a list of (feature: variant) combinations. For example, a mobile phone could be described as follows: \{(colour: red), (weight: 100 g), (connectivity: 3G), (battery capacity: 2000 mAh)\}. Mathematically, we formalise an artefact A as a set of (feature, variant) tuples of variable length, as presented in Equation (1). Each feature can only appear once in an artefact, and the number of variants per feature and the total number of features are limited, discrete, and strictly positive.

\[
A = \{(f_a, v_a), (f_b, v_b), \ldots | \forall i f_i \in \{1..f_{\text{max}}\}, \forall j v_j \in \{1..v_{\text{max}}\}, \forall a, b | a \neq b \rightarrow f_a \neq f_b \}
\]

(1)

Moreover, with Equations (2) and (3), we define ways to mathematically access artefacts. In Equation (2), we define \(v(f, A)\) as the variant of feature f in artefact A, if available. The variant “0” is reserved for “unknown feature”. In our example, the variant of the “colour” feature of our mobile phone is “red”, and the variant of the feature “sound system” is “0” because it was not yet defined.
\[ v(f,A) = \begin{cases} 0, & \exists f (f,v) \in A \\ v, & \exists f (f,v) \in A \end{cases} \]  

The set \( F(A) \) in Equation (3) represents the set of all known features in \( A \). In the mobile phone example, this would be the set \{colour, weight, connectivity, battery capacity\}.

\[ F(A) = \{ f_i \mid \forall v (f_i,v) \in A \} \]  

4.1.2. Effectual agent

We model effectuators as agents pursuing the task of artefact finalisation using their social network. Every effectual agent has an artefact, which is partially pre-populated with randomly drawn (feature: variant) combinations during the initialisation of the simulation. Although the literature states that effectual agents can “declare the effectual transformation complete and begin competing in alternative markets” (Sarasvathy & Dew, 2005a: 549), we focus on a single cycle of effectual artefact finalisation. Moreover, we assume that each stakeholder enters with sufficient resources to drive transformations and negotiations. Therefore, we do not explicitly model resources and potential constraints.

4.1.3. End customer

End customers represent the market demand in our current model. Formalised as agents, they fulfil the role of representing the conscious and sub-conscious market demand. The effectuation literature often notes the important role of end customers in examples, e.g., the customer-investors of U-Haul who helped to both shape and grow a one-man, one-truck company into America’s leading provider of moving and storage resources (Sarasvathy, 2001). More recent examples are the users of crowdfunding platforms, e.g., kickstarter. These customers represent the market demand to entrepreneurs who ask for their funding.

Thus, we formalise end customers as “passive” agents with two distinct properties: they can initiate the negotiation of pre-commitments upon approach and represent the market demand in
these negotiations. End customer agents represent the conscious market demand by an artefact that is used in negotiations. The sub-conscious demands are captured in a fitness landscape that enables end customer agents to assess the utility of a complete artefact.

We formalised the utility of an artefact using a well-known utility function known as the “fitness landscape” (Levinthal, 1997). The fitness landscape is a simple mathematical model that calculates a single fitness value for any given artefact, which represents the utility of an artefact for the market. Higher fitness values represent a higher market utility, i.e., a more desirable product for the market.

Similar to effectual agents, all end customer agents have an artefact representing their conscious demands. During the initialisation, the artefact is partially pre-populated with (feature: variant) combinations. Although the features are drawn randomly, the respective variant for an end customer agent is drawn to probabilistically represent the market utility. For instance, if the artefact represents a mobile phone, the utility of a high-capacity battery is considerably higher than the utility of a low-capacity battery. Therefore, the “high” variant is more likely to be assigned to the “battery capacity” feature of an artefact than the “low” variant.

4.1.4. Fitness landscape

We formalised the representation of market utility as an “NK-fitness landscape”, a well-known approach in management research employing simulation studies (Ganco & Agarwal, 2009; Gavetti & Levinthal, 2000; Levinthal, 1997). In accordance with effectuation theory, we implemented the simplest version of an NK-fitness landscape (K=0), which regards products as “near decomposable”, as required by theory (Sarasvathy, 2003).

The total utility U(A) of an artefact A in our fitness landscape is defined as the sum of the utility of its (feature: variant) combinations.
As indicated in Equation (4), we randomly initialise our utility function $u((f, v_i))$ for all possible (feature: variant) combinations. Moreover, we assign a value to features absent from an artefact. The fitness of an artefact representing the aforementioned mobile phone, for example, is calculated by awarding points to known features, e.g., “colour”, with high scores for “yellow” and low scores for “black”. Moreover, we assign points for missing features, e.g., “surround sound system”, which may be useless in business phones – and therefore score points by absence – but could be considered desirable for younger consumers. All points are then summed up to calculate the total utility value $U(A)$ of an artefact.

### 4.1.5. Social network

We initialise the network among effectual agents as well as between effectual and end-customer agents as an undirected free-scale network according to the generation procedures introduced by Barabási and Albert (Barabasi & Albert, 1999). Numerous studies have shown that free-scale networks best capture the structure of real-world networks (Aiello, Chung, & Lu, 2001; Barabasi & Albert, 1999; Broder et al., 2000; Jeong, Mason, Barabási, & Oltvai, 2001; Newman, 2000). Scale-free networks assume that the likelihood $p$ of having $k$ known contacts exhibits a negative exponential distribution ($p \sim k^{-\alpha}$).

The generation process of Barabási and Albert allows for a simple creation in two steps. First, the algorithm creates a network of effectual agents. Second, the algorithm attaches end-customer agents to the network. Initially, our algorithm creates a network of effectuators by cycling through all effectuators and creating exactly one tie originating from this effectual agent $i$. The target effectual agent $j$ of the tie is probabilistically chosen, as indicated in Equation (5).
\[ p(j) \sim \beta + \text{Degree}(j)^\alpha, \quad \alpha = 1, \beta = 0.1, \ i \neq j \quad (5) \]

The function \( \text{Degree}(j) \) refers to the number of contacts known by agent \( j \). Second, all end-customer agents are attached to the previously created network in the same manner.

### 4.2. Simulation mechanics

#### 4.2.1. Behaviours of effectual agents

Effectual agents can exhibit four behaviours to interact with their environment: approaching a known contact to present the current state of their artefacts, relaying information about received artefacts to known contacts, negotiating strategic alliances by exchanging pre-commitments, and inheriting the ideas of other effectual agents’ artefacts through transformation. These behaviours are controlled by the effectual agents’ persistence and docility. Both parameters are formalised as a variable representing low persistence and docility as 0.0 and high persistence and docility as 1.0.

#### 4.2.2. Approaching known contacts

Effectual agents can approach known contacts to present the current state of their artefacts. We formalised this behaviour as transferring copies of their artefact. For each contact, the effectual agent executes this behaviour probabilistically using the persistence parameter as the execution probability. The effectual agent can approach all known contacts – except for its stakeholders – once per turn. This behaviour is executed at the beginning of the simulation and every time its artefact changes. The recipient processes the proposed artefact in the next time step.

#### 4.2.3. Relaying received artefacts to known contacts

Effectual agents can relay information about other agents’ artefacts. We formalised this behaviour as probabilistically transferring copies of incoming artefacts to all known contacts except for stakeholders. This behaviour is triggered by every incoming artefact and is
probabilistically executed using the persistence parameter as the execution probability. Again, the recipient processes the relayed artefact in the next time step. To avoid multiple receptions of the same artefact, a list of previous recipients is transferred along with the artefact to decline the reception if it has been received previously.

4.2.4. Negotiation of pre-commitments

“Even the literature that is directly focused on negotiations has mostly neglected new venture creation processes” (Sarasvathy & Venkataraman, 2011). Despite the absence of a concrete algorithm, the literature (Sarasvathy & Dew, 2005a) implies multiple requirements for such an algorithm: (a) allow negotiation between two or more parties, (b) ensure that a position supported by a majority of participants is more likely to prevail, (c) support individual acceptance or refusal of negotiation results to ensure both self-selection into a project as well as the consideration of pre-commitments, and (d) negotiations are triggered by the approach of an effectual agent, as described in the “effectual cycle” (Dew et al., 2008).

As illustrated in Figure 2, we formalised the negotiation process as a simple, three-staged probabilistic negotiation. The behaviour is triggered upon the reception of an artefact. First, the algorithm determines all relevant stakeholders, i.e., project partners of the sending and receiving agents. The algorithm then gathers information on the artefacts of all involved agents. Second, the algorithm creates a combined list of occurring features and traverses it to create histograms of the frequency of the stakeholders’ preferred variants. Third, the algorithm creates a proposed negotiation result that includes all known features. The variant of each feature is determined probabilistically using the histograms as a (non-normalised) discrete probability mass function. For instance, if one stakeholder prefers the “colour” to be “red” and nine prefer it to be “blue”, the colour will be blue with a likelihood of 90%. Finally, the algorithm proposes the negotiation result to each stakeholder to receive feedback. A negotiation result is accepted only if all stakeholders agree on the adoption. In the event of an adoption, the algorithm replaces the
artefact of each stakeholder with the negotiation result and creates new network ties between all project partners.

After a negotiation result is created, each agent individually compares it to the state of its own artefact and decides whether to adopt it. As “[e]ffectuation assumes docility as a fundamental behavioural construct” (Dew et al., 2008: 498), we propose a docility-based formalisation of the acceptance process for effectual agents following a simple rule: a negotiation result that is more similar to the effectual agent’s own artefact will require less docility to be adopted.

We formalised this rule as a calculation of the share of conflicting features in relation to the total number of features held by the artefact of the deciding effectual agent. A conflicting feature is a different variant of the same property, e.g., a “red shell” instead of a “blue shell”. We assume that as the share of conflicted over known features for an effectuator increases, an effectuator must be more docile to accept these changes. Therefore, in our formal representation, an effectuator e with artefact A accepts the negotiation result R if the relative share of conflicting features is lower than its docility as indicated in Equation (6).

$$\frac{\left|v(f_i, A) \neq v(f_i, R) \mid f_i \in F(A)\right|}{|F(A)|} \leq \text{docility}$$

(6)

4.2.5. Inheriting features through transformation

As presented in the literature review, we formalise artefact transformation as a single abstract behaviour formalised as follows: the process is triggered by an unsuccessful negotiation. The approached agent now compiles a list of features that are exclusive to the proposed artefact. Then, it randomly chooses one of these features. Third, it includes the chosen (feature: variant) combination probabilistically in its artefact using its docility as the execution probability. Lastly,
the effectual agent initiates a negotiation for adoption with its stakeholders.

This formalisation implies the assumption that a certain amount of docility is required to transform an artefact based on external information. Moreover, it assumes that the adoption of a previously unknown feature is significantly more likely than the change of a known feature, which represents both the agent’s preferences and those of its stakeholders.

4.2.6. End customer agents’ access to the fitness landscape

During the negotiation, end customers can access the fitness landscape representing the market demand once. When confronted with the decision to adopt a negotiation proposal, end customer agents can access the fitness landscape to determine the total utility value of its artefact and the negotiation result. An end customer only accepts negotiation proposals with improved utility. The access restriction is essential to maintain the uncertainty state of the effectual environment. Knightian uncertainty is maintained because the number of design alternatives and their respective success probabilities cannot be determined by any agent. Goal ambiguity is maintained because end customers can only voice their conscious demands in negotiations but cannot determine the optimal variant of any feature. Finally, isotropy is maintained because the market demand for variants of known features cannot be estimated beyond the individual demand of the end customers. In conclusion, end-customer agents cannot search the fitness landscape to discover its optimum product.

4.2.7. Measures of network position

We focus on two widely recognized measures of centrality for social networks: degree centrality and betweenness centrality (Wasserman & Faust, 1994). Although both measures capture centrality, they emphasise slightly different aspects of this concept.

Degree centrality measures the number of known contacts of an agent. Therefore, degree centrality also represents the size of the agent’s personal network. For better comparability,
degree centrality is typically normalised by the total number of agents within a network.

**Betweenness centrality** captures the centrality of an actor by measuring its relevance for the information flow between any two other agents in the network. An agent is regarded as important for the information flow if it is part of the shortest route or “geodesic” between two agents. The measure is therefore defined as the number of appearances an agent makes on the geodesics of all pairs of other agents.

### 4.2.8. Measures of network shape

Burt (Burt, 1992) defines two key measures for the shape of an actor’s network: efficiency and constraint.

**Efficiency** is the degree to which the ties of an agent are non-redundant. Thus, in a highly efficient network, an agent is acquainted with contacts that do not know each other but are well connected themselves. In a network with low efficiency, all known contacts only provide access to agents that can also be reached through other ties as well, rendering the network largely redundant. Burt argues that entrepreneurs benefit from a highly efficient network. The exact measure is given in the literature (Burt, 1992: 53).

**Constraint** is a measure that captures the extent to which acquainted contacts impede the exploitation of structural holes. Whereas efficiency measures the extent to which direct ties are redundant, constraint measures the degree to which contacts are acquainted. Burt proposes that the acquaintance between known contacts – the absence of a structural hole between them – impedes the application of arbitrage and competition for information. Therefore, a low constraint level indicates an abundance of structural holes resulting in many opportunities for arbitrage and competition for information or resources.

Burt defines $C_i$ as the measure of constraint for actor $i$, as indicated in Equation (7). The term $p_{ij}$
represents the relative share of energy and time actor i invested in the relationship with j. For simplicity, we assume that actors divide their energy among their known contacts equally.

\[ C_i = \sum_{i \neq j} \left( p_{ij} + \sum_{q \neq i,j} p_{iq}p_{qj} \right)^2 \]  

(7)

As indicated in Figure 3, the constraint is lowest for a star-shaped network around an actor and highest for a complete network. Moreover, the constraint measure depends strongly on the inverse of known contacts. As our subsequent analysis compares actors with largely differing numbers of contacts, we normalised Burt’s constraint measure by multiplying it by actor degree. This normalisation results in a continuous value range between 1 (star-shaped network) and 4 (complete network).

4.2.9. Measurement of performance: demand satisfaction

The investigation on performance requires the definition of a performance measure. We define performance analogously to traditional performance measures (Narver, Slater, & MacLachlan, 2004; Vorhies & Morgan, 2005) as the similarity of supply and demand and refer to it as demand satisfaction. The more similar an effectual agent’s artefact is to those of an end customer agent, the higher the score. The demand satisfaction \( DS(i) \) of an agent \( i \) is calculated as a sum over all individual comparisons of an effectual agent’s artefact with all end customer agents’ artefacts. As indicated in Equation (8). The individual comparison score \( ds(A_i, A_j) \) is calculated as the share of met conscious demands of end customer \( j \) by effectuator \( i \).

\[ DS(i) = \sum_{j \neq i} ds(A_i, A_j) \]

\[ ds(A_i, A_j) = \frac{|\{v(f, A_i) = v(f, A_j) \mid f \in F(A_j)\}|}{|F(A_j)|} \]  

(8)
5. Results

To investigate the development of social networks of entrepreneurs using control-based logic over time as well as the performance effects of centrality and structural holes, we conducted a series of simulation experiments. We simulated an environment consisting of 40 end customer agents and 20 effectual agents. All agents received a randomly created artefact consisting of 5 (feature: variant) combinations drawn from 10 available features and 2 respective variants. The fitness landscape and network were randomly initialised for each run. We used values of 0.6 for persistence and 0.2 for docility. To gain a sufficiently large dataset, we repeated the simulation 1,200 times. Each run consisted of 40 turns, which was sufficient to reach a stable state. In each turn, all aforementioned measures regarding performance, centrality, and structural holes were computed for all agents. In total, we collected 1.44 GB of data that we subsequently analysed to investigate our research questions.

5.1. Performance impact of network position

We conducted two analyses to clarify the impact of network position on performance. First, we correlated demand satisfaction, degree, and betweenness centrality at the end of the simulation in a stable state. Second, we compared the initial degree and betweenness centrality to the final value of demand satisfaction for each effectuator.

For the first analysis, we extracted the final values of degree and betweenness centrality, demand satisfaction from the data. In Figures 4a and 4b, we display the average level of demand satisfaction per level of centrality. To account for the continuous nature of betweenness centrality, we grouped the values into 20 possible levels of betweenness centrality.

----------------------------------
Insert Figures 4 a-c here
----------------------------------
The results clearly indicate a connection of degree centrality with demand satisfaction. Starting with an average demand satisfaction of 0.4 for a degree centrality of 0.0, demand satisfaction increases considerably up to 0.8 at level 0.4. Then, the speed of growth decreases. Demand satisfaction increases to 1.0 for a degree centrality of 1.0. The correlation of betweenness centrality and demand satisfaction reveals a decrease of demand satisfaction with increasing betweenness centrality. Whereas low betweenness centrality is associated with a median demand satisfaction of 0.7-0.8, increasing betweenness centrality to 0.8 reduces demand satisfaction to 0.4-0.5. Moreover, the relationship between betweenness centrality and demand satisfaction is significantly more distorted than the aforementioned relationships.

Our second analysis concerned the impact of initial network configuration and final demand satisfaction. Thus, we analysed the impact of initial degree and betweenness centrality on the resulting demand satisfaction at a stable simulation state, which was always reached in turn 40. To ensure the availability of data over the entire range of initial centrality, we focused on a single agent, which was initialised with a given number of known contacts covering the complete range from 1 to 59.

As presented in Figures 5a and 5b, both initial degree and betweenness centrality have an impact on final demand satisfaction. Furthermore, in both cases, demand satisfaction increases strongly with centrality from 0.0 to 0.3-0.4 up to 0.7-0.8, after which demand satisfaction grows only marginally with increasing initial centrality to approximately 0.9 for both centrality measures.
5.2. Performance impact of network shape

Our second series of analyses targeted the performance implications of structural holes as defined by Burt (1992). Thus, we investigated the effects of constraint and network efficiency. For the analysis of efficiency and demand satisfaction, we grouped the continuous values into 20 levels of efficiency and 60 levels of normalised constraint to obtain meaningful results. As shown in Figure 6a, demand satisfaction decreases with efficiency from approximately 0.9 for minimal efficiency values to 0.3 for maximally efficient networks. As shown in Figure 6b, demand satisfaction increases gradually with normalised constraint from approximately 0.4 for low normalised constraint (1.0) to approximately 0.5 for a normalised constraint level of 2.5. Then, demand satisfaction increases considerably to 1.0 for the highest possible level of normalised constraint (4.0).

-----------------------------
Insert Figures 6a-b here
-----------------------------

5.3. Development of social networks over time

Our third series of analyses targeted the development of the previously presented measures over time. To better understand the dynamics of the development of effectual social networks, we tracked the development of degree, and betweenness centrality as well as efficiency and constraint for each time step of each simulation run. Figures 7 and 8 capture the average values of the aforementioned measures per time step.

As reflected in Figures 7a and b, over time, the average degree centrality increases from an initial value of approximately 0.05 to 0.35 at time step 18. The average betweenness centrality decreases from an initial value of 0.03 to 0.015 at time step 18. All values are constant after this time, indicating the stabilisation of the simulation.
Figures 8a and 8b indicate that over time, average efficiency decreases from its initial value of approximately 0.3 to 0.15 at time step 18. The average normalised constraint increases from an initial level of 1.0 to 2.5-3.0 at time step 18.

6. Discussion and implications

Although the creation and utilisation of social networks are critical for entrepreneurs, the underlying theory regarding entrepreneurial social networks is still underdeveloped. Moreover, “classical” networking strategies are at odds with control-based entrepreneurship approaches such as effectuation. Focusing on the impact of network structure, we used an agent-based computer simulation of effectuation to understand how entrepreneurs using control-based logics benefit from network position and shape. Moreover, we analysed how these entrepreneurs re-organise their social networks to increase performance.

6.1. Performance impact of network position

Our simulation experiments regarding network position reveal a significant performance impact of the network position in general and the initial position in particular. Moreover, the analysis of development over time reveals that control-based entrepreneurship actively improves the network position over time. However, these developments are captured to varying degrees by established centrality measures. Although degree centrality is positively correlated with demand satisfaction, betweenness centrality is negatively correlated with performance.

6.1.1. Performance implications of final network position
Our simulation experiments indicate a strong correlation between degree centrality and demand satisfaction at the end of the simulation, which is implicitly predicted by effectuation theory: instead of predicting the market demand, entrepreneurs using control-based logic shape their market by actively including stakeholders in the creation of their artefacts. In turn, these joint artefacts meet the demand of the stakeholders, i.e., project members. Consequently, the increasing size of the stakeholder group drives both demand satisfaction and degree centrality.

Degree centrality can be easily measured empirically and has been shown to have a positive impact on entrepreneurial venture performance before. However, the underlying reasons were different: for “classical” entrepreneurship high degree centrality increases the likelihood to acquire resources, such as market information or capital. For control-based entrepreneurship on the other hand, degree centrality is a proxy for successful control of the demand landscape. The difference in underlying reasons can best be captured with betweenness centrality, a measure that cannot easily be obtained empirically because a nearly complete disclosure of the network is necessary. Betweenness centrality captures the extent to which an actor serves as an “information hub” within a network. The role of the information hub is positively associated with performance in general entrepreneurship theory. Granovetter refers to the “strength of weak ties” (Granovetter, 1973), Burt’s “tertius gaudens” strategy relies on being an information hub, and several studies underline the importance of access to diverse information (Baum et al., 2000; Burt, 2004; Krackhardt, 1995; Zaheer & McEvily, 1999). Thus, high betweenness centrality should result in high venture performance. However, our simulation clearly reveals that the opposite is true for control-based entrepreneurship: successful entrepreneurs using control-based logic re-organise their network through the inclusion of actors and thereby “short circuit” the path that is necessary for information flow between unacquainted project members. In this manner, actors increase their degree centrality while decreasing their betweenness centrality. This explanation is supported by the development of degree and betweenness centralities over
time. Successful entrepreneurs using control-based logic manage to re-organise their networks by increasing degree centrality and decreasing betweenness centrality through their partnering approach.

6.1.2. Performance impact of initial network position

An important topic raised by practitioners is the initial impact of network position on future venture performance. Thus, we analysed the impact of initial degree and betweenness centrality on the final level of demand satisfaction. Although the general positive relationship between degree centrality and demand satisfaction holds, there is an important difference: our simulation reveals that beyond a certain threshold, additional centrality adds only infinitesimal levels of demand satisfaction for the agent. This result implies that a certain number of contacts is required to optimally apply control-based networking methods, but beyond a certain level, the initial “connectedness” of an entrepreneur using control-based logic does not significantly contribute to performance.

Of particular interest is the counter-intuitive relationship between initial betweenness centrality and future performance, i.e., demand satisfaction. Although negatively correlated for stable environments, initial betweenness centrality is positively correlated with final demand satisfaction in our simulation. In conjunction with the development of betweenness centrality over time, this paradox can be explained by the fact that an initially high betweenness centrality constitutes a potential opportunity to acquire a diverse set of stakeholders. Entrepreneurs using control-based logic can only increase demand satisfaction if they manage to successfully realise this potential through collaboration and subsequent introduction, i.e., the creation of a direct network path. This behaviour, in turn, decreases betweenness centrality.

6.2. Performance impact of network shape

In addition to the effects of network position, theory regarding the performance impact of
network shape for control-based entrepreneurship logics, such as effectuation, is still underdeveloped. As discussed in the literature review, Burt (1992) and Obstfeld (2005) have proposed strategies for entrepreneurs to benefit from certain network shapes. Based on Simmel’s concept of “tertius gaudens” (1950), Burt developed an entrepreneurship approach that effectively employs networks rich in structural holes by actively keeping network contacts separate and benefits from brokering between these contacts. In 2005, Obstfeld proposed an opposing strategy called “tertius iungens”, which focuses on connecting actors in a social network. We contribute to these streams of research by affiliating control-based entrepreneurial logics, such as effectuation, with these tertius iungens strategy and by providing a performance analysis in a highly uncertain environment.

6.2.1. Performance impact of efficiency

The negative correlation between final demand satisfaction and efficiency indicates that highly efficient networks are not suitable for control-based entrepreneurship logics, such as effectuation. Moreover, control-based entrepreneurship logic actively reduces efficiency by introducing unacquainted new stakeholders. In effect, a network with low efficiency indicates that a control-based entrepreneur succeeded to gather stakeholders and to shape market demand in his or her favour.

These findings initially seem contradictory to Burt’s approach to entrepreneurial networking that deems efficient networks favourable for entrepreneurs. However, a closer analysis of Burt’s implicit assumptions reveals the reasons for these unexpected results. Burt’s tertius gaudens strategy proposes to identify good ideas stemming from known contacts and to combine them in innovative ways (Burt, 2004). However, this approach implies that 1) the environment allows for the identification of good ideas and that 2) a transactional relationship with network contacts is sufficient. These conditions are unfulfilled in environments suitable for control-based entrepreneurship: effectuation, for example, assumes an entrepreneurial design space dominated
by Knightian uncertainty, Marchian goal ambiguity, and isotropy. It is thus impossible to assess the quality of an idea in this environment. Moreover, control-based entrepreneurship requires full commitment and continuous involvement of each stakeholder; thus a transactional relationship is insufficient. Therefore, control-based entrepreneurs simply cannot employ the advantages of an efficient network.

6.2.2. Performance impact of normalised constraint and structural holes

In our simulation, normalised constraint exhibits a positive relationship with demand satisfaction. Moreover, our longitudinal analysis reveals that Entrepreneurs using control-based logic actively re-organise their network to increase normalised constraint. Again, these results seem to contradict Burt’s structural holes theory that proposes a negative impact of highly constrained networks. Once more, the juxtaposition of the tertius gaudens strategy and effectuation as an example of control-based entrepreneurship reveals the cause: transactional relationships with stakeholders as well as the application of arbitrage and brokering are not part of the effectual toolbox. Thus, effectuation has no means to profit from structural holes. Entrepreneurs using control-based logic rather actively include new potential stakeholders to decrease the number of structural holes around them. Although entrepreneurs using control-based logic thereby constrain themselves, they also constrain their partners and bind them to their projects.

This is also reflected in the literature: Sarasvathy and Dew, for example, opposes the compatibility of the tertius gaudens strategy and effectuation (Sarasvathy & Dew, 2005a, 2005b). Given the necessity for co-creation and joint action, control-based entrepreneurship logics, such as effectuation, are conceptually closer to Obstfeld’s tertius iungens (Obstfeld, 2005).

In addition to the inability to profit from structural holes, control-based entrepreneurship logics even suffer from them: in his analysis of constraint and “good ideas” (2004), Burt reveals that networks with low constraint enable entrepreneurs to generate superior ideas yet fall short to
implement them. As noted by Obstfeld (2005), action toward implementation is taken by entrepreneurs with rather constrained networks. Thus, an “action-based” (Brettel et al., 2012) approach such as effectuation, requires constrained networks to be successful.

In effect, both efficiency and normalised constraint reveal that control-based entrepreneurship is a successful implementation of the tertius iungens strategy that actively re-organises networks to jointly shape market demand. While Burt’s proposed strategy may be incompatible with control-based entrepreneurship logics, the proposed measures can successfully be applied to identify suitable network shapes for both approaches.

6.3. Limitations and avenues for further research

Our study has several limitations that signal avenues for further research which can be grouped into three categories: development of effectuation theory, empirical validation, and enhancement of our effectuation model.

Our formalisation of effectuation contributes to a more precise definition of the effectual process and the entities involved. However, the current literature does not sufficiently examine the network-related aspects of effectuation. Further development of effectual networking theory requires additional case studies regarding the details of the networking process. Such studies should focus on negotiation yet include the termination of projects and the exclusion of stakeholders as well as a more detailed understanding of resource acquisition, distribution, and application.

Empirical validation of our theoretical propositions is required to either support our findings or challenge and thereby improve our formal model. Moreover, a validation of the practical applicability of our normalised constraint measure could validate our findings and contribute to theory development.
Finally, our formal model provides a multitude of improvement opportunities. First, the inclusion of resources would enable access to a vast field of research opportunities. Second, the inclusion of competing strategies would foster the investigation on conditions for normative superiority. This way, entrepreneurs could make an informed choice regarding their (networking) strategy.
References


7. Appendices

Figure 1: Based on (Dew et al., 2008; Sarasvathy & Dew, 2005a)

Figure 2: 

<table>
<thead>
<tr>
<th>Feature</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>v1</th>
<th>v2</th>
<th>Proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

*Taken from my paper “Individual vs. Collective Control in Effectual Social Networking”*
Figure 3:

\[ p_{ij} = \frac{1}{\text{Degree}(i)} \]

\[ p_{aj} = 0 \]

\[ C_i = \sum_{ij} \left( p_{ij} + \sum_{q \neq i,j} p_{iq} p_{qj} \right)^2 \]

\[ = \sum_{ij} \left( \frac{1}{\text{Degree}(i)} + \sum_{q \neq i,j} p_{iq} \cdot 0 \right)^2 \]

\[ = \sum_{ij} \frac{1}{(\text{Degree}(i))^2} \]

\[ = \text{Degree}(i) \cdot \frac{1}{(\text{Degree}(i))^2} \]

\[ = \frac{1}{\text{Degree}(i)} \]

\[ p_{ij} = p_{aj} = \frac{1}{\text{Degree}(i)} \]

\[ C_i = \sum_{ij} \left( p_{ij} + \sum_{q \neq i,j} p_{iq} p_{qj} \right)^2 \]

\[ = \sum_{ij} \left( \frac{1}{\text{Degree}(i)} + \sum_{q \neq i,j} \frac{1}{(\text{Degree}(i))^2} \right)^2 \]

\[ = \sum_{ij} \left( \frac{1}{\text{Degree}(i)} + \frac{\text{Degree}(i) - 1}{(\text{Degree}(i))^2} \right)^2 \]

\[ = \sum_{ij} \left( \frac{2}{\text{Degree}(i)} - \frac{1}{(\text{Degree}(i))^2} \right)^2 \]

\[ = \sum_{ij} \left( \frac{2}{\text{Degree}(i)} \right)^2 \]

\[ = \text{Degree}(i) \cdot \frac{4}{(\text{Degree}(i))^2} \]

\[ = \frac{4}{\text{Degree}(i)} \]
Figure 4a, 4b:

Final demand satisfaction and degree centrality

Impact of initial degree centrality

Final demand satisfaction and betweenness centrality

Impact of initial betweenness centrality
Figure 6a, b:

Final demand satisfaction and efficiency

Final demand satisfaction and normalized constraint

Figure 7a, 7b:

Degree centrality over time

Betweenness centrality over time
Figure 8a, b:

Average efficiency over time

Normalized constraint over time
Out of Thin Air? Simulating Entrepreneurial Opportunity Creation and the Impact of Environmental Complexity and Stakeholder Behavior

Abstract

In 2012, leading entrepreneurship researches reviewed the progress made on the development of entrepreneurship into “a science of the artificial”. They revealed significant need for further discoveries in three areas: the nature of opportunities, the processes applied by entrepreneurs and the interplay of opportunities, entrepreneurs and the environment. Besides the definition of an opportunity and the way it comes into existence – whether it is created or discovered – Venkataraman et al. propose further research regarding entrepreneurial transformation mechanisms as the applied processes, their key parameters and their exact definition is still unclear. Moreover, they encouraged research regarding the individual-opportunity-nexus and the precise mechanism at work when entrepreneurs create/discover their businesses.

Our study contributes to the deeper understanding of these areas with an opportunity-centric computer simulation study on the interplay of opportunity, individual and stakeholders, and the environment. We develop a formal model of an entrepreneurial logic applied by expert serial entrepreneurs – effectuation – and use it to investigate the impact of environmental complexity on opportunity creation and the development of opportunities over time. We reveal the pivotal importance of docile stakeholder behavior and explain why it is critical for successful opportunity creation in complex environments. Moreover, we show that opportunity creation using effectuation follows an implicit, but fixed order of phases: core completion, attribute completion, and stakeholder group completion.

Keywords: opportunity creation, docility, entrepreneurship, effectuation, environmental complexity, agent-based simulation

1. Introduction

In 2012, leading entrepreneurship researchers (Venkataraman et al., 2012) discussed the progress made so far on developing entrepreneurship into a “science of the artificial” (Shane & Venkataraman, 2000), i.e. move beyond the mere recognition of causal links towards allowing to carefully tailor the entrepreneurial method to specific problem spaces. The fulfilment of this promise of “entrepreneurship as a science of the artificial” yet hinges on the deeper
understanding of three key research areas: the nature of opportunities, the transformation methods applied by entrepreneurs, and the interaction of opportunities and entrepreneurs (Venkataraman et al., 2012). As a consequence, a lively debate regarding the definition and origin of opportunities as well as the action and interaction of opportunities and individuals sparked among leading entrepreneurship researchers. While one side proposed the enactment of opportunities through entrepreneurs (Alvarez & Barney, 2013), the other proposed the existence of opportunities independent of them (Shane, 2012). Consequently, both groups also differ in the exact definition of an opportunity. With respect to the interaction of individual and opportunity, Venkataraman et al. encourage research beyond “simple and direct relationships […] between individual and opportunity” (Venkataraman et al., 2012, p. 28). Proposing a thorough investigation on the interplay of opportunity, individual, and their context (Shane, 2012) – a call in line with other leading researchers (Sarasvathy & Venkataraman, 2011) as well. Proposing a shift from person-centric research towards a process-based view on entrepreneurship, Venkataraman et al. emphasize the research on transformational techniques as well. We therefore used effectuation, an entrepreneurial logic used by expert serial entrepreneurs, as a procedural basis for our studies. Effectuation is a non-predictive logic that strives to control future market development by shaping it. It relies on co-creation, collaboration and self-selection in order to jointly create a market (Dew et al., 2008; Sarasvathy & Dew, 2005a). Initial studies indicate a positive effect of these practices on venture performance (Read, Song, et al., 2009).

Given the intertwined, complex longitudinal processes at work, we use computer simulation as a method to investigate on opportunity creation and the impact of environmental complexity and entrepreneurial behaviour as proposed by leading simulation researchers (Davis et al., 2007).

With our study, we want to contribute to the deeper understanding of opportunity creation and the interplay of opportunities, entrepreneurs, and the environment they are interacting in, by answering two research questions: (1) How does environmental complexity influence the
creation of opportunities by entrepreneurs? (2) How do opportunities evolve over time and to what extent is opportunity evolution influenced by stakeholder behaviour?

2. Literature review

Based on a conceptual framework by Shane & Venkataraman (2000), the ideas regarding an “entrepreneurial method”, and the idea to see entrepreneurship as a “science of the artificial” (Sarasvathy, 2003, p. 203), Venkataraman et al. (2012) revealed significant demand for further research in three areas: the nature of opportunities, entrepreneurial transformation mechanisms, and the interaction of opportunities and entrepreneurs. Answering their call, we consequently provide an overview on these areas starting with the nature of opportunity and the nexus of individual and opportunity including the mechanisms of interaction termed “action-interaction nexus” (Venkataraman et al., 2012, p. 28). We conclude this chapter with a detailed review of a decision logic preferred by expert serial entrepreneurs – effectuation, as if forms the basis of our computer simulation.

2.1. The nature of opportunities

Initially, literature conceptualized two distinctive ways of how opportunities come into existence: opportunities exist and are discovered by entrepreneurs or opportunities are created by entrepreneurs (Alvarez & Barney, 2007; Steyaert & Hjorth, 2003). While the former premise implies exogenously induced shocks as basis for entrepreneurial action, the latter proposes an endogenous impulse as basis for the creation of an opportunity. Initially seen as disjoint concepts, recent research joined both approaches and proposed a co-evolvement (Sarason et al., 2006) of opportunity, entrepreneur and stakeholders (Dutta & Crossan, 2005; Gartner, 1994). Klein put it very beautifully as “opportunities are best characterized neither as discovered nor created, but imagined” as “the concept of opportunity imagination emphasizes that gains (and losses) to not come into being, objectively, until entrepreneurial action is complete” (Klein, 2008,
p. 12). Venkataraman et al. (2012) conclude that “opportunities in the world have to be made through the actions and interactions of stakeholders […] using materials and concepts found in the world”, hinting that the world or environment as well plays a role as well. While a generally accepted terminology is yet to be adopted (Alvarez & Barney, 2013), a preliminary review of the concepts of opportunities being “made” or “created” on one side and “found” or “discovered” on the other revealed highly similar – to the extent of identical – meanings in both cases. Those rejoinders however, reveal a certain amount of vagueness in the definition of the term “opportunity”: if opportunities – much like the laws of physics – exist independently of the entrepreneur they cannot be made at the same time. Shane addressed this conundrum by splitting the “opportunity” in two parts: the “opportunity” as a metaphysical conceptualization that exists independently of the entrepreneur, i.e., the possibility to profitably transport passengers by means of air travel, and the “business plan” as the entrepreneurial implementation of an opportunity, whose specifics are negotiated by entrepreneurs and stakeholders (Shane, 2012, p. 15). While being criticised as “not testable” by Alvarez and Barney (Alvarez & Barney, 2013), the distinction of “opportunity” and “business plan” seems to provide a much needed clarification of the opportunity concept. It does however depend on a compatible concept of the market as subsequently discussed.

Given the emphasis on creation and transformation of the effectuation approach, we conclude that the completion of the artifact as addressed in the former chapter rather refers to the “business plan”-aspect of an opportunity. However, a holistic implementation of the entrepreneurial process requires a metaphysical “opportunity”-aspect as well. We implemented this aspect as a “fitness landscape” as subsequently described in the “formal modelling” section.

2.2. The interaction of opportunities and entrepreneurs

Informed by Shane’s formulation of an “individual-opportunity nexus” (Shane, 2004) (earlier: Shane & Venkataraman, 2000), Venkataraman et al. (2012) sketch out the second area of
research as the investigation on a link between characteristics of entrepreneurs, their actions and interactions and the opportunities they pursue. Initially, Shane and Venkataraman (Shane & Venkataraman, 2000; Shane, 2004) proposed that while individual and opportunity are existing independently of each other, a special connection between these two entities was necessary enable the individual to transform this opportunity into reality (See fig. 1a). In further iterations, informed by the call for the study of inter-subjectivity (Sarasvathy & Venkataraman, 2011), and based on findings regarding entrepreneurial characteristics and actions and their impact on the pursued opportunities (Baker, Gedajlovic, & Lubatkin, 2005; Dencker, Gruber, & Shah, 2009), a deeper understanding of the action and interaction of opportunities on one side and entrepreneurs and stakeholders on the other was proposed as a research field in 2012 (Venkataraman et al., 2012) (See fig. 1b). Venkataraman and Sarasvathy referred to it as “Action-Interaction nexus”. Moreover, they ascertain that “an opportunity [– not as defined by Shane –] consists of at least three things:[…] objective person-opportunity nexus, […]subjective interpretation of objective data, […] inter-subjective basis for a market” (Venkataraman et al., 2012, p. 26). In light of Alvarez and Barneys criticism that while “these “objective” conditions exist, […] this] does not deny that entrepreneurs sometimes enact the opportunities they intend to exploit” (Alvarez & Barney, 2013, p. 6), we propose to extend the action-interaction nexus further to include the market as well. While an opportunity can exist independently from an individual, it cannot exist independently of a market. Going back to the airline example, another individual could have taken Richard Branson’s place, but the founding of virgin airlines (or a similar airline) could not have happened without a mass of customers willing to use the airline. Moreover, it required generations of (involuntary) entrepreneurs preparing the market by fighting the notion that “If God had meant us to fly he’d have given us wings”. In conclusion, both literature stream arguing for the creation and the discovery have merit, but it requires a holistic perspective (such as an action-interaction-market nexus, see fig. 1c) to join and understand them. In this study, we take a first step and use a joint model of opportunities, entrepreneurs, stakeholders, and end customers
to research the effect of environmental (i.e. market) complexity.

---------------------------
Insert figure 1 here
---------------------------

2.3. Entrepreneurial transformation mechanisms

Venkataraman et al. (2012), ascertaining that recently discovered mechanisms applied by entrepreneurs transcend the Schumpeterian idea of new combinations and emphasize modification techniques that transform opportunities beyond their inherent boundaries. Building on Dew, Read, Sarasvathy & Wiltbank (Dew et al., 2010) and Goodman (1978) they reason that while combinations of ideas within their boundaries can lead to incremental innovation, entrepreneurs use exaptation to achieve technological and economic innovations beyond evolitional developments. Venkataraman et al. (2012, p. Table 1) list 11 mechanisms used in the interaction of opportunities and entrepreneurs. Roughly half of these can – to a certain extent – be directly or indirectly associated with effectuation. A deeper understanding of effectuation is therefore helpful for the development of entrepreneurship as a science of the artificial.

Effectuation is a control-based logic (Wiltbank & Dew, 2006) that aspires to shape future market developments rather than to forecast them. Effectuation relies on four principles: keeping the loss of any action affordable, using available means, treating contingencies as opportunities and relying on partnerships (Sarasvathy, 2001). Since its introduction in 2001, effectuation has received significant attention from the research community. The core concepts were reviewed and enhanced (Dew et al., 2008; Dew, Read, et al., 2009; Sarasvathy, 2004a) including the networking process (Dew et al., 2008; Sarasvathy & Dew, 2005a) and introduced to adjacent research (Chiles et al., 2008; Goel & Karri, 2006; Read, Song, et al., 2009). Moreover, quantitative studies revealed increased venture performance for business angels’ investments (Wiltbank et al., 2009), R&D projects (Brettel et al., 2012), and “new venture creation” (Chandler et al., 2011). However, “the actions and interaction of entrepreneurs and their
stakeholders” “could be a viable line of empirical research” (Venkataraman et al., 2012, p. 28).

The environmental preconditions for effectuation are characterized by three elements: Knightian uncertainty, Marchian goal ambiguity and isotropy (Dew et al., 2008). Knightian uncertainty (Knight, 1921) refers to an environment where both alternative implementations of an artifact – or an attribute thereof – and the probabilities for success are indeterminable. Marchian goal ambiguity (J. March, 1978) denotes the fact that neither the preferences of partners/stakeholder nor own preferences are completely known. Environmental isotropy “refers to the fact that in decisions and actions involving uncertain future consequences it is not always clear ex ante which pieces of information are worth paying attention to and which not” (Sarasvathy & Dew, 2005a).

2.4. Effectuation as a logic of expert entrepreneurs

Besides “Who I am” and “what I know”, networks (“whom I know”) are a key resource for effectuators as it provides both access to “new means” and “new goals” (Dew et al., 2008; Sarasvathy & Dew, 2005a). Despite the fact that “Inter-subjective interaction is the very essence of the effectual process” (Dew et al., 2008, p. 50), research on the network-related effects on the effectuation processes remains sparse.

As illustrated in figure 2 – based on Dew et al. (2008) – effectuation literature refers to four occasions that include or imply the use of networks. Firstly, effectuators approach known contacts and present their current artifact to initiate negotiations (Dew et al., 2008; Sarasvathy & Dew, 2005a). Secondly, effectuators can relay information regarding the artifact of other effectuators acting as a source of contingency (Sarasvathy, 2001). Thirdly, effectuators engage in negotiations regarding pre-commitments and partnerships (Dew et al., 2008; Sarasvathy & Dew, 2005a). Fourthly, effectuators transform their artifacts in multiple ways (Dew et al., 2010; Goldenberg, Lehmann, & Mazursky, 2001), triggered by external stimuli from their network.
2.4.1. **Presentation of artifacts**

In order to create or finalize an artifact, effectuators create a “project” (Dew et al., 2008, p. 50), approach known contacts and present their current artifact. While effectuation literature currently treats this action as a mere prelude to negotiation and transformation (Dew et al., 2008) it is an integral part of the market creation process. The persistence with which effectuators approach potential new stakeholders is a function of personality traits (“Who I am”) as well as the affordable loss constraint as networking requires both time and effort.

2.4.2. **Relay of information**

While not explicitly mentioned in effectuation literature, this behaviour represents both a well-known way of information dissemination in networks (Burt, 1992; Granovetter, 1973) as well as an important factor of the source of contingency principle, which is a core dimension of effectuation (Sarasvathy, 2001).

2.4.3. **Negotiation in effectual networks**

In order to create or finalize an artifact, effectuators initiate negotiations with known contacts. These negotiations lead to commitments of a twofold nature: pre-commitments that specify the nature of the emerging artifact as well as resource commitments to drive its finalization. Given that the market for the emerging artifact is only emerging as well, the negotiations will mostly focus “on the characteristics of the project […] rather than the type and quantity of returns” (Dew et al., 2008, p. 50) of it. Over time, the increase in number of stakeholders creates “two concurrent dynamic cycles” (Sarasvathy & Dew, 2005a, p. 1) leading to an increase in committed resources as well as the concretization of the joint artifact.

Literature on the behaviour of effectuators during negotiations has determined that the docility of
an effectuator is “a fundamental behavioural construct” (Dew et al., 2008, p. 50). The concept of docility was introduced by Simon (1990) and portrayed as “the tendency to depend on suggestions, recommendations, persuasion, and information obtained through social channels as a major basis for choice“ (Simon, 1993b, p. 156). Simon later clarified that, “We are highly susceptible to social influence and persuasion, susceptibility that I will call docility. I use the term ‘docility’ here in its sense of teachability or educatability – not in its alterative sense of passivity or meekness” (Simon, 1997).

2.4.4. Transformation of artifacts

New market creation through effectuation is not restricted to the negotiation of pre-commitments with stakeholders. As a result of a case study, Dew et al. (Dew et al., 2010) listed 9 types of transformation that expert serial entrepreneurs apply to their artifacts in order to create new markets. Contrary to negotiations, this action does not lead to a contribution of new contacts or resources.

For a formalized model of effectual networking, we focused on the types of transformation that can be triggered by stimuli from networks. These transformation types (“deletion and supplementation”, “composition and decomposition”) share a common theme: effectuators create a new artifact through transformation based on their old artifact as well as ideas provided by an externally induced stimulus. As the network is a key resource of information, we argue that the transformation can be triggered by the approach of other effectuators in search for potential stakeholders.

We therefore include an abstract version of the aforementioned transformation type in our formal model and refer to it as “pick-up”. Moreover, we argue that the execution of this behaviour depends on the personality traits of effectuators, namely their docility, as “influence and persuasion” (Simon, 1997) is exercised in the presentation of artifacts, even if it does not lead to
successful negotiations of a joint project.

3. Agent-based simulation as a method

We employ agent-based simulation to examine our research questions. This “third way of doing science” (Axelrod, 2003) allows for unique insights in areas where theory is at an intermediate stage and complex, intertwined processes are at work (Davis et al., 2007). Simulation overcomes the inherent complexity limitations of empirical research (Harrison et al., 2007) and allows gathering data on all aggregation levels beyond empirical feasibility (McKelvey, 2004). The derived propositions will in turn help to design efficient studies for empirical verification.

While simulation as a method is still underrepresented in management science, “several influential research efforts have used simulation as their primary method” (Harrison et al., 2007, p. 480). Organizational learning (J. G. March, 1991) and adaptation (Levinthal, 1997) problems as well as the balance of different kinds of search processes (Gavetti & Levinthal, 2000; Rivkin & Siggelkow, 2003) and innovation diffusion (Abrahamson & Rosenkopf, 1997) benefited from simulation. More recent examples include strategy (Davis et al., 2009) and characteristics of a company (Ganco & Agarwal, 2009).

Following a well-known roadmap (Davis et al., 2007), we developed a comprehensive model of the effectual transformation process based on literature reviews and expert interviews. For implementation, verification, and validation of the model we used the acknowledged simulation framework Repast Simphony (North et al., 2007). This environment allows for the required longitudinal collection of data on agents and their stakeholders. Moreover, we employed Monte-Carlo simulation (Davis et al., 2009) to ensure statistically relevant results and conducted comprehensive sensitivity analyses and robustness checks to ensure the validity of our findings. For verification we used well-published empirical and theoretical studies (Brettel et al., 2012; Read, Dew, et al., 2009; Sarasvathy & Dew, 2005a; Wiltbank et al., 2009). Following the ideas
of Dew et al. (Dew et al., 2010) and Goodman (1978) we also included transformation mechanisms beyond stakeholder negotiation in an abstract form.

4. A formal model of effectual opportunity creation

Modelling effectual opportunity creation requires a formal representation of all relevant actors, objects, and processes of effectual opportunity creation. This includes effectuators and their artifacts as well as end customers both as potential stakeholders and as representation of the market the effectuators are creating. Effectuators and end customers form a network that allows effectuators to approach both end customers and other effectuators. According to literature (Dew & Sarasvathy, 2007), effectuators co-operate only with stakeholders – including end customers – that are willing to make pre-commitments. We paid special attention to the implementation of uncertainty as this is a major pre-requisite for effectuation (Sarasvathy, 2001).

In a nutshell, the simulation works as follows: starting in time step 1 all effectual agents start presenting their artifacts to known contacts. On approach, these contacts can then decide to engage in negotiations with the presenting effectual agent. If an agreement is reached, both stakeholders agree on a joint artifact and continue their search for additional stakeholders together. If no agreement is reached, the engaged party can however decide to pick up some new attributes from the presented artifact. Over time, the artifacts are refined and the simulation converges into a setting of multiple stable stakeholder groups. It is determined when it reaches a stable state, usually after 30-60 ticks.

---

9 Partly taken from my paper “Individual vs. Collective Control in Effectual Social Networking”
4.1. Objects and actors

4.1.1. Artifact

The artifact refers to a product, service, project or goal an effectuator is finalizing (Sarasvathy, 2001). Following the product description of an arbitrary merchandise, e.g., a mobile phone, we formalized an artifact as a list of (attribute: variant)-tuples. A mobile phone could for example be described as: \{\text{(colour: red), (weight: 100gr), (connectivity: 3G), (battery capacity: 700mAh)}\}. Mathematically speaking, we formalize an artifact \(X\) as a set of (attribute, variant)-tuples of variable length as presented in equation (1).

\[
X = \{(a_i, v_i), (a_j, v_j), \ldots | \forall k a_k \in \{1, \ldots, a_{\text{max}}\}, \forall i v_i \in \{1, \ldots, v_{\text{max}}\}, \forall i, j \in \{1, \ldots, a_{\text{max}}\} a_i \neq a_j \}
\]

\[
a_{\text{max}} \in \mathbb{N}^+, v_{\text{max}} \in \mathbb{N}^+
\]

Moreover, with equation (2) and (3) we define ways to mathematically access an artifact.

Function \(v(a, X)\) retrieves the employed variant \(v\) of an attribute \(a\) in artifact \(X\), if available. The set \(A(X)\) represents the set of all known attributes \(a_i\) in \(X\), e.g., colour, weight or connectivity.

\[
v(a, X) = \begin{cases} 0, & \neg \exists a_i (a_i, v_i) \in X \land a = a_i \\ v, & \exists a_i (a_i, v_i) \in X \land a = a_i \end{cases}
\]

\[
A(X) = \{a_i | \forall v_i (a_i, v_i) \in X\}
\]

4.1.2. Effectuator

We modelled effectuators as agents pursuing the task of artifact finalization using their network. Every effectual agent has an artifact, which is partially pre-populated with randomly drawn (attribute: variant)-tuples during the initialization of the simulation. To focus on network-related behaviour, we simplified available theory in two points: extant literature implicates that effectual agents can “declare the effectual transformation complete and begin competing in alternative markets” (Sarasvathy & Dew, 2005a, p. 549). In contrast, we only study a single cycle of
effectual artifact finalization. Moreover, we assume that each stakeholder enters sufficient resources to drive transformations and negotiations. Therefore we omit an explicit modelling of resources and potential constraints.

4.1.3. **End customer**

End customers represent the market demand in our current model. Formalized as an active part of our model, they fulfil the important role of providing insights into what customers want and regard as useful. Effectuation literature mentions the important role of end customers often in examples, e.g., the customer-investors of U-Haul, that helped to both shape and grow a one-man one-truck company into Americas leading provider of “moving and storage resource” (Sarasvathy, 2001). More recent examples are the users of crowd funding platforms, e.g., kickstarter. These customers are representing the market demand to the entrepreneurs that ask for their funding. In part, this role is also assumed by the effectuators and their prior knowledge “What I know” themselves.

We therefore formalize end customers as “passive” effectuators with two distinct properties: they can initiate the negotiation of pre-commitments upon reception of an artifact and represent the market demand including “subconscious desires” in these negotiations by being able to assess the differences in “usefulness” of two given artifacts.

We formalized the usefulness using a well-known approach known as “fitness landscape” (Levinthal, 1997). The fitness landscape is a simple mathematical model that calculates a single “fitness value” for any given artifact, which represents the usefulness of an artifact for the market. Higher fitness values represent a more desirable product for the market.

Like effectual agents, all end customer agents have an artifact representing their conscious demands. During the initialization it is partially pre-populated with (attribute: variant)-tuples. While the attributes are drawn randomly, the respective variant is drawn to probabilistically
represent the market demand. If, e.g., the artifact represents a life vest, the utility of a yellow-coloured vest is much higher than of a black-coloured one. Therefore the “yellow”-variant is much more likely assigned to the colour attribute of an artifact than the “black” variant.

4.1.4. Fitness landscape

We formalized the representation of usefulness as an “NK-fitness landscape”, a well-known approach in management research employing simulation studies (Ganco & Agarwal, 2009; Gavetti & Levinthal, 2000; Levinthal, 1997). In accordance with effectuation theory, we implemented the simplest version of an NK-fitness landscape (K=0), which regards products as “near decomposable“ as demanded by theory (Sarasvathy, 2003).

The fitness of an artifact X on our fitness landscape is defined in equation (4) as

\[
FIT(X) = \sum_{a_i \in A(X)} \text{fit}(a_i, v_i) + \sum_{a_i \notin A(X)} \text{fit}(a_i, 0)
\]

\[\forall a_i \in \{1, ..., a_{\text{max}}\}, v_i \in \{0, ..., v_{\text{max}}\} \text{ fit}(a_i, v_i) \in [0,1] \text{ randomly} \tag{4}\]

As indicated in equation 4, we randomly initialize our fitness function fit((f_i, v_i)) for all possible (attribute: variant)-tuples. Moreover, we also assign a value to attributes not available in the artifact. The fitness of an artifact representing the aforementioned life-vest would for example be calculated by awarding points to known attributes, e.g., “colour” with high scores for “yellow” and low scores for “black”. Moreover we would also assign points for missing attributes, e.g., “entertainment system”, which might be useless in emergency situations – and therefore scoring points by absence – but could be considered extravagant for snorkelling trips. All points are then summed up to calculate the total fitness FIT(A) of an artifact.

4.1.5. Network

We initialize the network among effectual agents as well as between effectual and end customer agents as an undirected free-scale network according to generation procedures introduced by
Barabási and Albert (Barabasi & Albert, 1999). Numerous studies showed that free-scale networks best capture the structure of real-world networks (Aiello et al., 2001; Barabasi & Albert, 1999; Broder et al., 2000; Jeong et al., 2001; Newman, 2000) and the generation process of Barabási and Albert allow for a simple creation in two steps. Firstly, the algorithm creates a network of effectual agents, secondly the algorithm attaches end customer agents to the network. Our algorithm creates a network of effectuators by cycling through all effectuators and creating exactly one tie originating from this effectual agent i. The target agent j of the tie is probabilistically chosen as indicated in equation (5):

\[ p(j) \sim \beta + Degree(j)^\alpha, \quad \alpha = 1, \beta = 0.1, \ i \neq j \]  

The function Degree(j) refers to the number of contacts the agent j knows. Secondly, all end customer agents are attached to the previously created network the same way.

4.2. Simulation mechanics

4.2.1. Behaviours of effectual agents

An effectual agent can exhibit four behaviours in order to interact with their environment: approaching known contact in order to present the current state of his or her artifact, relaying information on artifacts of other effectual agents to known contacts, negotiating strategic alliances by exchanging pre-commitments, and pick up ideas of other effectual agents’ artifacts through transformation. These behaviours are controlled by the effectual agents’ persistence and docility. Both constructs are formalized as a variable representing low persistence and docility as 0.0 and high persistence and docility as 1.0.

4.2.2. Approaching known contacts

Effectual agents can approach known contacts in order to present the current state of their artifact. We formalized this as passing on copies of the artifact. For each contact the effectual
agent executes this behaviour probabilistically using the persistence parameter as execution probability. The effectual agent can approach all known contacts – except for its stakeholders – once per turn. This behaviour is executed in the beginning of the simulation and every time its artifact changes. The recipient processes the proposed artifact in the next time step.

4.2.3. **Relaying received artifacts to known contacts**

Effectual agents can relay information on other agents’ artifacts. We formalized this as probabilistically passing on unaltered copies of incoming artifacts to all known contacts, except stakeholders. The behaviour is triggered by every incoming artifact and is probabilistically executed using the persistence parameter as execution probability. Again, the recipient processes the relayed artifact in the next time step. In order to avoid multiple receptions of the same artifact, a list of previous recipients is passed along with the artifact to decline the reception if it had been received before.

4.2.4. **Negotiation of pre-commitments**

“Even the literature that is directly focused on negotiations has mostly neglected new venture creation processes” (Sarasvathy & Venkataraman, 2011). In spite of the absence of a concrete algorithm, literature (Sarasvathy & Dew, 2005a) implies multiple requirements for such a formalization: (a) allow the negotiation between two or more parties, (b) ensure that a position carried by a majority of participants is more likely to prevail (c) support individual acceptance or refusal of negotiation results to both ensure self-selection into a project as well as consideration of pre-commitments, and (d) negotiations are triggered by the approach of an effectual agent as described in the “effectual cycle” (Dew et al., 2008).

As illustrated in figure 3, we formalized the negotiation process as a simple, three-staged probabilistic negotiation. Firstly, the algorithm determines all relevant stakeholders, i.e., project partners of the sending and receiving agent. Subsequently, the algorithm gathers information on
the artifacts of all involved agents. Secondly, the algorithm creates a joint list of occurring
attributes and traverses it to create histograms of the frequency of the stakeholders’ preferred
variants. Thirdly, the algorithm creates a proposed negotiation result that includes all known
attributes. The variant of each attribute is determined probabilistically using the histograms as a
non-normalized discrete probability mass function. If, e.g., 1 stakeholder prefers the “colour” to
be “red” and 9 prefer it to be “blue” the artifact will be blue with a likelihood of 90%. Lastly, the
algorithms proposed the negotiation result to each stakeholder to receive feedback. A negotiation
result is accepted only if all stakeholders agree on the adoption. In the event of an adoption, the
algorithm replaces the artifact of each stakeholder with the negotiation result and creates new
network ties between all project partners.

---------------------------
Insert figure 3 here
---------------------------

After a negotiation result is created, each effectuator will individually compare it to the state of
his or her own artifact and decide on the adoption. As “[e]ffectuation assumes docility as a
fundamental behavioural construct” (Dew et al., 2008, p. 498), we propose a docility-based
formalization of the acceptance process following a simple rule: the more similar the negotiation
result is to the effectual agent’s own artifact, the less docility is required to adopt it.

We formalized this rule as calculation of the share “conflicting” attributes in relation to the total
number of attributes the artifact of the deciding effectual agent holds. A conflicting attribute is a
different variant of the same property, e.g., a “red shell” instead of a “blue shell” or a “large
battery pack” instead of a “small battery pack”. We assume that the larger the share of conflicting
over known attributes is for an effectuator, the more docile an effectuator has to be in order to
accept these changes. Therefore in our formal representation, an effectual agent accepts a
negotiation result, if the relative share of conflicting attributes is lower than its docility. If an
effectuator i with artifact X examines the negotiation result R, it can accept this result if equation
(6) is true.

\[
\frac{\left| \{ v(a_i, X) \neq v(a_i, R) \mid a_i \in A(X) \} \right|}{|A(X)|} \leq \text{docility}_i
\]  

4.2.5. Pick-up of attributes

As presented in the literature review, we formalize artifact transformation as a single abstract behaviour that we refer to as “pick-up”. It is formalized as follows: the process is triggered by a previously unsuccessful negotiation with the owner of the proposed artifact. Initially, the algorithm compiles a list of attributes that are exclusive to the proposed artifact. Secondly, it randomly chooses one of these attributes. Thirdly, it includes the chosen attribute probabilistically into the receiving effectuator agent’s artifact using the effectual agent’s docility as execution probability. Lastly, the receiving agent initiates a negotiation for adoption with its stakeholders.

This formalization implies the assumptions that a certain amount of docility is required to transform an artifact based on external information. Moreover, we assume that the adoption of a previously unknown attribute is significantly more likely than a change of a known attribute, which in all likelihood represents both the agent’s and its stakeholders’ preferences.

4.2.6. End customer agents’ access to fitness landscape

During the negotiation, end customers can access the fitness landscape representing the market demand just once. When confronted with the decision to adopt a negotiation proposal, end customer agents can interrogate the fitness landscape whether the proposal as a whole is meeting both their conscious and subconscious demands better than their current artifact. This restricted access formalizes the notion that “customers do not know what they want until they see it” which has been discussed as goal ambiguity. The access restriction is essential to maintain the state of the effectual environment. Knightian uncertainty is maintained as the number of design
alternatives and their respective success probabilities cannot be determined by the end customer agent. Goal ambiguity is maintained as end customers do not provide goals such as desirable attributes as they cannot search the fitness landscape to determine how desirable an attribute is. Lastly, isotropy is maintained as the market demand for variants of known attributes cannot be estimated beyond the individual demand of the end customers. In conclusion, end customer agents cannot search the fitness landscape to discover its optimum.

4.2.7. Measurement of performance: demand satisfaction

The simulation environment calculates the degree of demand satisfaction (DS) of each effectual agent in each time step. The Measure is defined analogous to traditional performance measures (Narver et al., 2004; Vorhies & Morgan, 2005) taking into consideration all potential end customers. The closer an effectual agent’s demand vector is to an end customer agent’s vector, the higher the performance score. The total score $DS(i)$ of an agent $i$ is calculated as a sum over all comparisons with all end customer agents $j$ as indicated in equation (8). The individual comparison $ds(X_i, X_j)$ is calculated as the share of met demands of end customer $j$ by effectuator $i$.

$$DS(i) = \sum_j ds(X_i, X_j)$$
$$ds(X_i, X_j) = \frac{|\{v(a, X_i) = v(a, X_j) \mid a \in A(X_j)\}|}{|A(X_j)|}$$  \quad (8)

5. Results

To investigate on the opportunity creation mechanisms of effectuation, we conducted a series of simulation experiments. Our simulation consisted of 30 effectual agents and 60 end customer agents. Their product vectors were randomly initialized with 5 (attribute, variant)-tuples. Moreover, we changed the level of collective docility between 0.1 and 0.95. The fitness

---

10 Partly taken from my paper “Entrepreneurial Mingling Secrets”
landscape was initialized randomly. We gathered a total of ~12.7 GB of data running our simulation 500 times for each parameter combination.

5.1. Impact of environmental complexity

To answer our first research question regarding the impact of environmental complexity on effectual opportunity creation, we conducted two series of experiments. In both experiments we analysed the impact of one environmental complexity parameter as well as the level of collective docility on performance. We used the average level of demand satisfaction of all effectual agents as performance measure. We did exclude those agents that were not able to find any stakeholders.

5.1.1. Impact of number of attributes ($a_{\text{max}}$)

As reflected in figure 4, the level of demand differs between ~0.6 and 1.0 depending on the level of collective docility. With increasing number of attributes $a_{\text{max}}$, these levels stay mostly constant.

5.1.2. Impact of number of variants ($v_{\text{max}}$)

As displayed in figure 5, our second series of experiments was concerned with the variation of the number of variants $v_{\text{max}}$. With respect to demand satisfaction, with only one existing variant of all attributes ($v_{\text{max}} = 1$) the average satisfaction is 1.0 regardless of the level of collective docility. With increasing number of variants, the level of demand satisfaction changes considerably. For high values of collective docility ($\geq 0.6$) the level of demand satisfaction stays at 1.0 or decreases only slightly down to ~0.7. For lower values of collective docility ($\leq 0.5$) the level of demand satisfaction decreases significantly down to ~0.1 – 0.3.
5.2. Evolution of opportunities over time

To investigate on our second research question regarding the development of opportunities over time, we observed the time to complete different “stages” of opportunity development over time. We tracked the share of agents that had finished a certain step of the opportunity development process. These steps are a) finalization of variant selection, i.e., when do variants of existing attributes stop changing, b) completion of attribute acquisition, i.e., when is the list of attributes complete, and c) completion of stakeholder group, i.e., when is the last stakeholder added to the group? We performed analyses for different levels of collective docility and persistence to detect possible influences of these key behaviour parameters.

As reflected in figure 6, there is a clear order in which effectual agents complete these stages: firstly, agents finalize the variants of existing attributes, secondly, they complete their attribute sets and thirdly complete their stakeholder group. While figure 6 is a graphical representation of our results for a collective docility level of 0.2 and persistence level of 0.6, we provide a comprehensive overview of stage completion times in table 1 for all combinations of collective docility and persistence levels. As reflected in table 3, after 3-6 ticks all agents have finalized the variants of available attributes. After 3-6 ticks, all agents have finalized their attribute set. Lastly, after 6-11 ticks all stakeholder groups are complete. While the lengths of the phases vary significantly, the order is always kept.
6. Implications

In 2012 leading entrepreneurship researchers corroborated their goal to develop entrepreneurship into a “science of the artificial” (Venkataraman et al., 2012). They laid out three key research areas that required further substantiation in order to reach this goal: the constitution of opportunities, entrepreneurial transformation methods, and the “person-opportunity nexus” (Venkataraman et al., 2012, p. 26). With our simulation-based research on entrepreneurial opportunity creation, we contribute to a deeper understanding of the interplay of opportunity and entrepreneur, namely the impact of environmental complexity and behavioural parameters of entrepreneurial communities. Moreover, we provide new insights into the development of opportunities over time. We used the effectuation process – a decision logic applied by expert serial entrepreneurs – as an instantiation of entrepreneurship. While the theory is still in a “nascent/intermediate” state (Perry et al., 2012, p. 11) it is regarded as an important building block of entrepreneurship (see Venkataraman et al., 2012, p. Table 1) and provides a sufficiently concrete process description. Given the nascent state of effectuation theory and the amount of required data from different actors and points in time, we chose computer simulation as our research method. Simulation is well suited for the improvement of “underdeveloped theory” (Davis et al., 2007, p. 482) and is “particularly valuable when the theory seeks to explain longitudinal and processual phenomena that are challenging to study using empirical methods because of their time and data demands” (Davis et al., 2007, p. 495).

6.1. Impact of environmental complexity

Our simulation experiments regarding environmental complexity reveal a stark difference regarding different types of complexity. While the number of existing attributes has little to no impact on demand satisfaction across all levels of collective docility, the number of existing variants strongly affects demand satisfaction. Moreover, the impact of existing variants depends strongly on the level of collective docility.
6.1.1. Impact of the number of existing attributes

The number of existing attributes has little impact on both performance measures. Aside from some initial fluctuation the performance is solely depending on the level of collective docility. While environmental complexity is usually associated with negative impact on performance for start-ups (Aldrich, 1990; Brittain & Freeman, 1981), our simulations provides a partially contrary result. A close review of the effectuation process and our implementation however reveals the reasons for the “missing” decay in performance: our implementation of the negotiation process favours the inclusion of additional attributes. Without a resource constraint or the assumption of bounded reality, the inclusion of new attributes does not have any negative impact. Moreover, the active involvement of stakeholders, especially end customers, ensures that attributes with negative impact on the products usability (fitness) are not included. Ultimately, with respect to new attributes, our implementation of effectuation is not constrained by adverse effects of low docility. We argue that affection for innovation is a core quality of an entrepreneur so unless there is a good reason to refuse a new idea, it is included. We acknowledge however, that the research on effectual stakeholder negotiation is not yet conclusive enough and propose especially this part of effectuation as an avenue for further research. This detail of the effectual negotiation process also explains the slight decrease in demand satisfaction for, e.g., 6 existing attributes: The product vector of each agent is initialized with 5 attributes. For environments with only few more existing attributes, the result of a negotiation is likely an agent’s own product vector complemented with one new attribute. The likelihood of this new attribute being inferior to not having it at all is 1/3. Therefore the result is more likely to be turned down by end customer agents. For larger numbers of existing attributes the number of attributes added at once by a negotiation is likely larger, in sum rather positive, and therefore more likely to be acceptable for end customer agents.

With respect to docility, a level of ≥0.5 delivers superior performance while lower levels deliver
inferior results. The positive impact of docility is in line with entrepreneurship literature. Initially theorized by Simon (1990), leading effectuation researchers (Dew et al., 2008; Sarasvathy & Dew, 2005a) regard it as crucial for the effectuation process. Prior simulation results (Jansen, 2013a) also support the importance of collective docility.

In conclusion, a “positivist attitude” towards new ideas, i.e. attributes, enables effectuators to incorporate large numbers of new product attributes without risking the loss of performance. They require however the input of external information via stakeholders, i.e., end customer agents, to prevent group thinking and ensure real product improvement.

6.1.2. Impact of the number of existing variants

Contrary to the number of existing attributes, the number of existing variants – per attribute – has significant impact on both demand satisfaction. While the demand satisfaction in the previous experiments only changed marginally at a level of 30 existing attributes (allowing $2^{30} \approx 1.07e9$ combinations), at a comparable value of 8 existing variants (allowing $8^{10} = 2^{30} \approx 1.07e9$ combinations as well) the value of demand satisfaction has already changed significantly. Moreover, the impact of existing variants is moderated by the level of collective docility. While those environments with a level of collective docility $\geq 0.6$ shows minor to no impact of the number of variants, environments with a collective docility $< 0.6$ reveal a significantly negative impact of increased levels of existing variants.

We therefore propose that collective docility has a moderating effect with respect to environmental complexity: For high levels, environmental complexity has little to no negative impact on effectual performance, while for low levels environmental complexity has a significantly negative effect on effectual performance. In our experiments, the breaking point of docility was between 0.5 and 0.6. As there is no empirical measure of docility available\textsuperscript{11}, the

\textsuperscript{11} Unless one counts in cattle (Burrow et al., 1988)
interpretation of this value requires both further experimentation and empirical validation.

These surprising results regarding collective docility warrant a reinterpretation of this parameter. Current literature on effectuation theory (Dew et al., 2008; Sarasvathy & Dew, 2005a) treats docility merely as a necessity. Prior simulation experiments (Jansen, 2013a) revealed positive impacts. In light of these results however, there seems to be a lower bound for collective docility that needs to be met especially in complex environments in order to ensure successful opportunity development. With respect to the “individual-opportunity nexus” (Sarason et al., 2006; Shane & Venkataraman, 2000; Shane, 2004; Venkataraman et al., 2012) the implementation of an opportunity (not its existence) is conditional on the availability of suitable stakeholders/partners. Coming back to the "virgin airline”-example of Venkataraman et al. (2012, p. 29): the deal required both Branson as individual and Boeing as a docile stakeholder to implement the opportunity now known as virgin airlines. Given our results, we propose that the docility of both actual and potential stakeholders is a critical element of the mechanics of this nexus, depicted as “action-interaction-nexus” (Venkataraman et al., 2012, p. 28). These results also adds an interesting element to the analysis of the “inter-subjective” as demanded by leading effectuation researchers (Sarasvathy & Venkataraman, 2011, p. 125).

6.2. Evolution of opportunities over time

In our last series of experiments we investigated on the evolution of opportunities over time. Our findings reveal a stable order or “phases” of finalization the opportunity undergoes in the effectual process: firstly, the variants of known attributes are finalized, i.e. fixed (core completion). Secondly, the incorporation of new attributes is finalized (attribute completion) before lastly, the inclusion of additional stakeholders is finalized (stakeholder group completion). In numerous experiments under varying degrees of both collective docility and persistence the time differences in finalization varied, but never changed their order.
This fact is noteworthy for two key reasons: this order is neither predefined by – usually actor-centric – effectuation process literature (Dew et al., 2008; Sarasvathy & Dew, 2005a) nor even mentioned in the on-going discussion on created vs. discovered opportunities (Alvarez & Barney, 2007; Dutta & Crossan, 2005; Gartner, 1994; Sarason et al., 2006; Steyaert & Hjorth, 2003). Moreover, it is contrary to the causation-based idea of entrepreneurship to rather start with target customers and define the core attributes of the product or service accordingly (Sarasvathy, 2003, p. Fig. 2).

According to literature, “[o]pportunities emerge as a function of means”(Dew et al., 2008, p. 48) and they do so in a “usually path dependent fashion” (Sarasvathy, 2003, p. 214). Especially in environments with high docility this suggests that attributes and their respective variants and the involved stakeholders can change at any point in time. Again, a detailed review of the effectual negotiation process and the idea of docility reveal the reasons for the emergence this counterintuitive order in opportunity finalization. Simon (1997) defined docility as “teachability or educatability – not in its alterative sense of passivity or meekness”. We therefore argue that during a negotiation highly docile stakeholders are more willing to accept a contrary position, yet they are not without an opinion on it. Taken the iterative effectuation process into account, this leads to an opportunity development along the lines of our results: Initially, in 1-to-1 negotiations, the negotiation power between incumbent and potential stakeholders is even. Therefore the negotiation result is likely to contain conflicting variants for known attributes for both parties. In order to accept these, sufficient docility is required on both sides. In subsequent negotiations, the balance of power shifts: The incumbent stakeholders outnumber the new potential stakeholders and even while the incumbent stakeholders may be highly docile, it becomes more and more unlikely that a negotiation result will change the variant of an established attribute. From an effectuation theory perspective this can be regarded as the exercise of control. While changing the variant of an established attribute gets more and more unlikely,
stakeholder groups are still open for new/unknown attributes. As long as these do not conflict with established ones or lead to an inferior outcome there is no opposition to them and they can easily be included into the opportunity. After a while however, all relevant, i.e. existing, attributes are included. At this point in time, additional stakeholders are still welcome to join a project; however, it is very unlikely that they will be able to influence the opportunity. Given the overwhelmingly skewed balance of negotiation power, joining a project becomes a “take it or leave it” decision.

In effect, our opportunity-centric research approach revealed two key aspects of effectuation: docility is more than a mere “free parameter”, it is paramount when acting in a complex environment. Moreover, timing is of the essence as the evolution of opportunities follows its own schedule: key attributes of opportunities are usually implemented in the early phases of development and are unlikely to be changed, regardless of the docility of the involved stakeholders.

6.3. Limitations and further research

The presented study holds several limitations that pose avenues for further research in the following areas: empirical validation, scale development, theory development, and model enhancement.

All our propositions are derived from a computer simulation based on current literature on the effectuation processes. Therefore, our results require empirical validation to both verify the results as well as the theory they were built on. We hope that the derived propositions will foster empirical research to reason their hypotheses efficiently and reduce the amount of work as they now “know where to look”.

As docility plays a crucial role in this and our prior paper (Jansen, 2013a) as well as in key process descriptions of effectuation, the development of a docility scale is imperative to
understand and validate the mechanisms at work in effectuation.

With respect to theory development, the deeper understanding of the effectual negotiation process, tactics and influential parameters is paramount. Being declared a “hot topic” of effectuation research\textsuperscript{12}, we see the understanding of effectual inter-subjective processes as a key building block for further theoretical and computer-simulation-based studies.

Lastly, our simulation model holds important limitations that provide further avenues for research. Using all available process descriptions available, we realize the shortcomings – besides negotiation – especially with respect to resources. Future simulation work should include a resource-based model to analyse the impact of the affordable loss principle and the interplay of resource constraints and the effectual process.

\textsuperscript{12} By Sarasvathy during a talk at the “Entrepreneurship Research and Teaching Conference 2013”, Lyon
References


Appendices

Individual-opportunity  action-interaction  action-interaction-market

Figure 1a,b,c: development of nexuses and proposal of an action-interaction-market nexus

Based on (Dew et al., 2008; Sarasvathy & Dew, 2005a)

Figure 2: Effectuation process including transformation

Gather artifacts of all relevant stakeholders  Create histogram of occurring variants  Create negotiation proposal

<table>
<thead>
<tr>
<th>Feature</th>
<th>Artifacts</th>
<th>Variants</th>
<th>Proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1 A2 A3 A4</td>
<td>v1 v2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2 2 1 1</td>
<td>2 2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2 2 1 1</td>
<td>2 2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2 2 2 1</td>
<td>1 3</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>- - - 1</td>
<td>1 0</td>
<td>1</td>
</tr>
</tbody>
</table>

13 Taken from my paper “Individual vs. Collective Control in Effectual Social Networking”
Figure 3: Example of probabilistic negotiation algorithm

Figure 4: Impact of number of existing attributes on demand satisfaction
Figure 5: Impact of the number of variants on demand satisfaction

Figure 6: Finalization stages of opportunities over time
<table>
<thead>
<tr>
<th>Time [ticks]</th>
<th>Level of collective docility</th>
<th>Level of collective persistence</th>
<th>Variant changes complete Attribute acquisition complete</th>
<th>Stakeholder group complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

Table1: Changes in phase completion time