IRPact: An Actor-centered framework for Innovation Diffusion in Integrated Resource Planning

Simon Johanning[1], **Fabian Scheller**[1,2], **Thomas Bruckner**[1]

[1] *Institute for Infrastructure and Resources Management (IIRM), Leipzig University*
[2] *Energy Systems Analysis, Division of Sustainability, Department of Technology, Management and Economics, Technical University of Denmark (DTU)*
*Correspondence should be addressed to *johanning@wifa.uni-leipzig.de*

Journal of Artificial Societies and Social Simulation xx(x) x, (20xx). Doi: 10.18564/jasss.xxxx
Url: http://jasss.soc.surrey.ac.uk/xx/x/x.html

Received: dd-mmm-yyyy Accepted: dd-mmm-yyyy Published: dd-mmm-yyyy

Abstract: Insights into the diffusion process can help decision makers to detect weak points of potential business models. Yet, due to the multitude of factors to consider, modeling the diffusion of innovations is a very challenging task. In the literature, various models and methodologies to address this problem can be found. Among these, empirically grounded agent-based modeling turns out to be one of the most promising approaches. However, the current culture is dominated by papers that fail to document critical methodological details. Thus, existing agent-based models for real-world analysis differ extensively in their design and grounding and therefore also in their predictions and conclusions. Being aware of this, this research paper seeks to identify requirements as building blocks in order to design and develop a versatile, but robust model to assess innovation diffusion processes. Subsequently, a formal approach is developed based on the derived model entities, dynamics and foundations. The main objective of this modeling approach is to achieve modularity and flexibility, as well as clarity through an explicit description of the concepts used. This is achieved by a three-layer approach, with a super agent layer, an agent layer and a sub-agent layer. Building on this, an object-oriented code base, organized in 24 packages and 273 classes is created. This empirically grounded agent-based modeling framework can be utilized by innovation diffusion researchers in order to build upon existing frameworks and concepts and to model a diverse range of domains in innovation diffusion.

Keywords: Agent-Based Modeling, Agent-Based Product Diffusion Model, Innovation Diffusion, Human Decision-Making, Socio-Economic Simulation, Multi-Agent Simulation

Introductory remarks

Problem statement

0.1 Product or service innovation constitutes an effective means for organizations to create and maintain a competitive advantage. This is why it is important to understand how market actors engage with and adopt innovations, since even good innovations may fail or diffuse at a slow rate (Rogers 2003). For many companies, it is hard to predict how innovations will diffuse in a dynamic environment, resulting in uncertainty about whether an innovation is fit to become a sustainable business model.

0.2 This may be to a large part because the adoption of these innovations by intended target groups is not always assured, and as Frederiks et al. (2015) show, it does not solely depend on the qualities of the innovation. Instead, it takes place within a complex social system, in which the diffusion of the respective innovations depend on many factors and mechanisms (Schwarz 2007). Business models and innovations need to encompass the dynamics of the market setting by including the personal and mental structures of market participants. As Kiesling (2011) points out, *"[...] the diffusion of innovation paradigm postulates that markets are in fact dominated by social influences [...]."*

0.3 Thus, decision makers responsible for these innovations are confronted with making informed decisions about complex matters (Kiesling et al. 2012). Insights into the diffusion of innovations can help to detect weak points of potential business models and innovation marketing. Particularly quantitative models of innovation diffusion analysis that account for the complexity of the modeled system might assist decision makers in the development of effective strategies. One promising approach for this is to employ **empirically grounded Agent-Based Models (eABMs)** (Bonabeau 2002; Macal & North 2010; Kiesling et al. 2012). An agent-based model (ABM) is a model in which entities are modeled individually as autonomous, social, reactive and proactive agents (Wooldridge 1998). Autonomous decision-making strategies in accordance with the agents' personal objectives describe the procedure of taking an action depending on several conditions. In order to model the strategies of the heterogeneous agents realistically, it is necessary to collect and analyze an extensive amount of empirical data to derive a theory for grounding (Glaser & Strauss 1967). In this sense, an agent needs to be theoretically and empirically grounded (Smajgl & Barreteau 2014a). In these models, **Innovation Diffusion (ID)** is understood as the analysis of the spread of an innovation (Rogers 2003).

0.4 In recent years, eABMs gained importance as a valuable methodology for describing diffusion processes (Kiesling et al. 2012). Thereby, they are particularly applied to reflect real-world market issues (Smajgl & Barreteau 2014b). *"In the spirit of modern complexity science, these models have the potential to reproduce and explain complex non-linear diffusion patterns observed in the real world as the result of relatively simple local micro-level interactions."* (Kiesling 2011). Despite the individual differences among models, many eABMs share a number of commonalities. However, as Bell et al. (2015) stated, in most cases models are developed primarily without regarding existing approaches and shared common structures, resulting in a lack of *"[...] a clear foundation of agreed-upon approaches and libraries that offer a baseline for problem solutions that characterize other modeling fields."* Providing a generic framework for ID assessment by integrating and extending modeling approaches of eABMs allows modelers to compare specific model mechanisms with little effort and to identify what kind of model would be most appropriate for the subject at hand.

Research Scope

0.5 This research paper aims to answer the following main research question: **"How can decision-making processes involving the adoption of sustainable products be assisted through a flexible, modular framework for eABMs on ID?"**. The major objective is to develop an integrated model framework suited to incorporate decisive components of proven models, as well as providing a formal model and a software implementation for existing and novel applications. It is directed at real-world case-based applications, as a variable decision support tool building on modern existing approaches in the literature. For a systematic process, this research intends to accomplish the following objectives:

- Review building blocks of eABMs for ID analysis in order to guide the model development process.
- Design a versatile eABM framework to evaluate ID processes under different system conditions.
- Implement a software implementation to facilitate the flexible configuration of the developed eABM.

From a scientific point of view, the presented eABM framework aims to analyze the ID process interrelations between customer behavior and incentive measures. In practice, the framework can help decision makers to evaluate business model innovations in a fast-changing environment and thus develop a sustainable business strategy.

Research Structure

0.6 This research paper is organized as follows: The section **Requirements Analysis** comprises the model requirements by reviewing and synthesizing existing modeling approaches. The developed eABM framework is outlined in the section **Framework Modeling**. The section **Software Implementation** gives insights into the software implementation. In the section **Concluding Remarks** the developed model is discussed and its contribution is sketched.

Requirements Analysis

0.7 eABMs are gaining importance as a valuable methodology for describing ID processes (Kiesling et al. 2012). Since these models are primarily applied to reflect real market issues, papers with real-world case studies to support decision makers are becoming more popular (Smajgl & Barreteau 2014b). Case-based applications *"have an empirical space-time circumscribed target domain."* (Boero & Squazzoni 2005). They are usually built *"to provide forecasts, decision support, and policy analysis [. . .]."* (Kiesling et al. 2012), showing that the application domain of eABMs for ID is very versatile. Similarly, existing eABMs differ extensively in their design and grounding and therefore also in their predictions and conclusions (Smajgl & Barreteau 2014b; Zsifkovits 2015).

0.8 Addressing this variety, this research paper builds upon a systematic review of eABMs as presented in (Scheller et al. 2018a,b). The major aim is to identify requirements in the form of building blocks in order to design and develop a versatile but robust model to assess ID processes. The derived **Model Entities** and **Model Dynamics** are discussed hereinafter. Moreover, aspects regarding the **Model Foundations** are outlined. An illustrative overview of the interplay of the identified and integrated model components is given in figure 1.

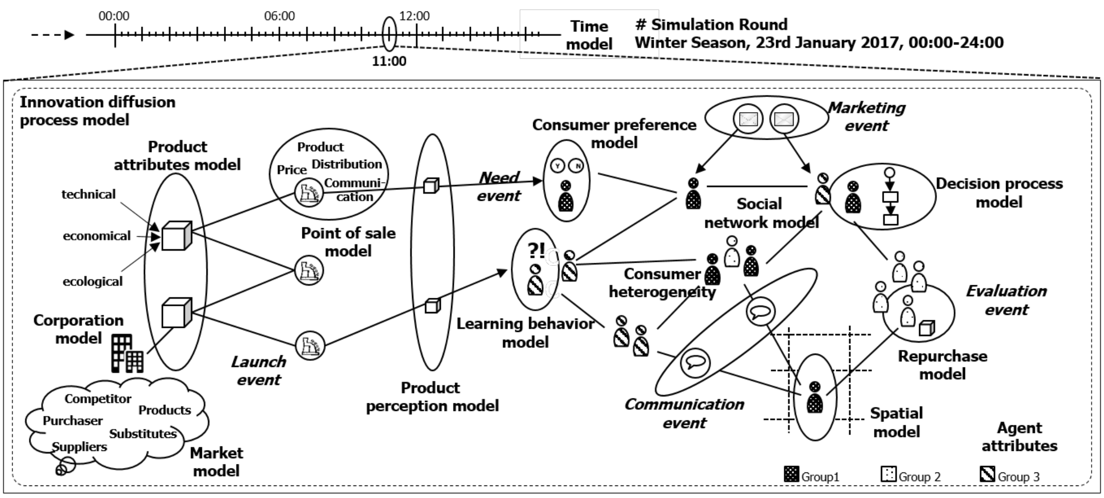

Figure 1: Illustrative interplay of identified model entities, dynamics and foundations.

Model Entities

0.9 Since innovation diffusion is primarily concerned with adoption decisions of actors, **consumer agents** (individuals or households) are generally the center of these models. Thereby, personal, economic and social attributes are used for their characterization. Examples for this are subjective norms (Graebig et al. 2014) or characteristics within the socio-economic coordinate system (Wolf et al. 2015), such as age, salary, years of ownership, average behavior (Eppstein et al. 2011), or innovativeness (Schramm et al. 2010). Adoption agents might also be heterogeneous with regards to their communication and decision behavior which are influenced by social and spatial factors (Schwarz 2007).

0.10 Actor heterogeneity can further be seen in aspects ranging from individual actor-centered ones up to those characterizing whole groups of agents and their relationship with other groups of agents. Thus, consumer agents can be seen from an individual actor-centered perspective or from a characterization of agents by types, encompassing, among others, behaviors, attitudes, social and communicative patterns, perception and decision approaches, often done on a socio-economic or psycho-sociological basis (such as sinus milieus, socio-economic groups, cognitive involvement in decision processes etc.). As such, in addition to modeling consumer agents, the ability of aggregating them in **groups** homogeneous enough to justify group identity (intra-homogeneity) and as distinct from other groups as possible (inter-heterogeneity) is important. Furthermore, flexibility in parameterizing these degrees of heterogeneity is important.

0.11 In addition to consumer agents, **company agents** have been considered in different eABMs. While Balbi et al. (2013) and Zhang & Nuttall (2011) only indirectly model them, Schramm et al. (2010) models corporate entities directly as brand agents. In this role, corporation entities also interact with the consumer agents to influence the behavior of other agents and should be able to take management decisions, carry out product management and engage in advertisement and marketing. This is even more so the case for models in which the point-of-sale (POS) is depicted as physically distinct **sales agent** with a range of products, product availabilities and prices, and a purchase process as implemented by Stummer et al. (2015). These agents may be locally bound or not embedded in the spatial model at all. Similar to other actors, a POS should exhibit proactive behavior and might take decisions, employ strategies, set the prices for the products and manage the availability of the products.

0.12 Additionally, **policy agents** are used both endogenously and exogenously in existing eABMs, e.g. in Zhang & Nuttall (2011) and Barreteau et al. (2014). While in the first case rules and regulations are set by authorities, in the second case administration tasks are aggregated by a policy agent. In order to depict this, a suitable framework should allow for the management of existing policies and the introduction of new ones, bundling regulatory aspects falling in the public sphere.

0.13 Obviously for ID models, the service or product innovations analyzed are of fundamental importance. Due to the heterogeneity of the investigated models, the technologies under investigation come in a number of flavors. With an abstract representation of **product models**, however, this diversity can be captured. Product attributes come in a large variety, ranging from technical parameters (Zhang & Nuttall 2011; Schwarz & Ernst 2009) through ecological characteristics (Windrum et al. 2009; Palmer et al. 2015) to cost-related parameters (Eppstein et al. 2011; Palmer et al. 2015). Since these are all characteristics potentially influencing the evaluation of a product, the various aspects can be abstracted into distinct **product attributes**. To enable the use of models sensitive to cognitive modeling, depicting not only the true qualities but also the **perception** of qualities is relevant. Perceptual aspects identified are ecological aspects (Eppstein et al. 2011), social perception (Stummer et al. 2015) or substitute availability (Windrum et al. 2009), which intends to model acquiring knowledge about qualities and existence of products. Perceptions are not static aspects though, as elaborated further below with the model dynamics.

Model Dynamics

0.14 Being an operative definition of newness, a source of consumer heterogeneity and an essential component of ID, **decision making** is arguably one of the most important aspects of ID modeling. It can be seen as the cognitive processes the actor employs for deciding on a product. Due to the multitude of aspects playing into the actor's decision, a number of other components need to interplay with it, such as the consumers' nature, available products and time. It further incorporates, for example, the channel described in Schramm et al. (2010), bringing together mass media and advertisement, communication, product attributes and decision processes. Social influence is another important aspect in decision making, as a *"person's perception that most people who are important to him think he should or should not perform the behavior in question"* (Graebig et al. 2014). Modeling the decision process is most commonly grounded in decision theories or in utilitarian approaches (van Eck et al. 2011; Palmer et al. 2015; Stummer et al. 2015; Schramm et al. 2010; Balbi et al. 2013; Rai & Robinson 2015). Additionally, Zhang & Nuttall (2011) use game theoretic approaches in their decision processes. On the contrary, Wolf et al. (2015) *"[. . .] model agent decision-making with artificial neural networks that account for the role of emotions in information processing".* While the kind of decision processes used can differ between consumer agents of heterogeneous groups (Schwarz 2007), within actor groups the same process should be used. The diversity of approaches shows that a decision process component needs to be highly flexible, and that all aspects non-essential to the decision process should be modeled in other components.

0.15 While products are mostly seen as static entities within the time frame of adoption, some models require some dynamics of the modeled products. This usually does not concern the attributes of products, but instead the relation of the entire product towards the simulation, as in Schwarz (2007), incorporating products that enter the simulation at a later time (**market introduction** of products) than others or become unavailable before the end of the simulation (**product discontinuation**).

0.16 To depict models sensitive to cognitive modeling, eABMs need to include incomplete information, cognitive distortions and (subjective) perception. Thus, they not only need to model the true qualities, but also the perception of qualities through a mechanism for determining the perceived value of a product attribute. This allows for modeling product attribute values for actors being based on perception instead of the true value of the product attribute. **Perception modeling** is seen among others in Stummer et al. (2015) with social perception or substitute availability, and is mentioned by Windrum et al. (2009), which intends to model acquiring knowledge about qualities and existence of products. In Wolf et al. (2015), the authors integrate mental representation into the interplay of communication, the social network, media, perception, and the decision model. The importance of the interplay between perceptions and the true nature of products can also be seen by Kiesling (2011), where the true attributes of the products can only be assessed with a **post-purchase evaluation** of the adopted products. Another aspect of incomplete information is the knowledge about the existence (**awareness**) of products.

0.17 Additionally, the difference in preferences is seen as another source of how heterogeneity is incorporated in eABMs. Preferences capture predilections and aversions of actors quantitatively, which often stand for (moral) values or goals, whereas product attributes are quantifiable properties of the product. Thereby, **preference modeling** of agent entities absorbs the categories of ecological aspects of products, such as environmental concerns (Palmer et al. 2015), as well as certain aspects of social perception (Stummer et al. 2015; Sopha et al. 2013). It further incorporates attitudes relating to preferences in order to be used by decision processes (Zhang & Nuttall 2011) or in other words to weigh product aspects (Eppstein et al. 2011). Thus, preference values need to be linked to product attributes in order to relate values to the evaluation of products.

0.18 As pointed out extensively before, the decision to adopt a product is seen as one of the most important aspects of agent-based ID models. The product adoption decision describes the process of purchasing a product by an actor adopting this product, fulfilling the needs corresponding to the product for the rest of the simulation or a limited product lifetime. Due to the large variety of decision procedures employed in existing models however, no uniform mechanism how agents are prompted to enter a decision process is named. A decision to adopt is always motivated by the non-fulfillment of some need of the actors to adopt, and how these needs arise and get evaluated towards should be caught in a component directed at **need modeling**.

0.19 Moreover, word of mouth is an important aspect of many models. Due to the multitude of possible communication contents, numerous model aspects fall into this category. Arguably the most important function of communication is to foster social perception (Stummer et al. 2015; Chappin & Afman 2013; Eppstein et al. 2011). Thus, the communication channels differ within the literature. One set of channels mentioned is connected to advertisement and mass media (van Eck et al. 2011; Stummer et al. 2015; Wolf et al. 2015). Others are communication channels between consumer agents, leading to it being subsumed by communication and the social network (Stummer et al. 2015; Rai & Robinson 2015; Wolf et al. 2015). The model described in Schramm et al. (2010) uses a more complex channel, which can be modeled by mass media and advertisement, communication, product attributes and the decision process. Since behavior and beliefs are mediated through communication within a social network, models exemplifying behavior and belief change (Wolf et al. 2015) also include communication. Further aspects derived from models that also touch upon communication are learning, personal attributes such as social influence and number of peers to communicate, social norms and an aspect of mental representation (Wolf et al. 2015). Due to the plurality of communication forms and contents, **communication modeling** thus needs to allow for heterogeneous and flexible schemes for communication between actors, as well as the communication processes. Communication needs to respect the social network of actors and communication schemes. A central aspect in communication is the manipulation of product perceptions of consumer agents.

0.20 Additionally, van Eck et al. (2011) imply a corporate entity using mass media. Stummer et al. (2015), Wolf et al. (2015) and Broekhuizen et al. (2011) make use of channels related to advertisement and mass media, while Schramm et al. (2010) model the interplay of mass media and aspects product attributes and decision processes. ID models should thus allow at least the change of the perception or preference of an agent, taking into account the true and desired nature of a product. Thus, more abstractly, an ID model needs to address **information and advertisement modeling**, particularly how a range of information and advertisement processes take place. This includes changing the perception or preference of an agent, taking into account the true nature of a product for information and some desired value for advertisement[1]. In order to allow company agents to make use of advertisement for their product portfolio, they need to be equipped with flexible advertisement facilities within the model.

Model Foundations

0.21 Even if only done implicitly, simulation models are always grounded on some modeling strategies related to (physical) fundamentals, in particular time, space and processes. The strategies defined here are important for the implementation as well as for processing logic. In particular, to depict temporal dynamics, time needs to be an integral part of ID models. With the exception of Kiesling (2011), **temporal models** are barely ever discussed explicitly, and the focus of many models is on process modeling, while temporal progress is only implicitly modeled. Despite the fact that almost all models employ discrete timing schemes (time passes in some temporal unit, and all or some agents act in each step), a more general framework should enable the implementation of different timing schemes (such as discrete and continuous). As the governor of temporal

[1]The major difference between information and advertisement is that information is actively sought out by consumer agents and describes how actors proactively seek out information within decision processes or other processes, whereas advertisement is targeted at them by company agents or exogenous events, and consequently the consumer agents are more passive.

dynamics, the temporal model needs to at least address how process events are handled within the temporal context.

0.22 Since the temporal aspect of many mechanics depends on the processes modeled, many temporal aspects of the model are situated within the **process model** and the temporal model is described slenderly. The process model usually (at least implicitly) governs the stages of the innovation decision process (with at least the stages of awareness, trial and adoption) of Rogers (2003), and specifies the execution of processes (for discrete temporal models) or actions between events (for continuous temporal models). It is thus subordinated to the temporal model, and adoption of products is governed by the process model. Generally, product adoption is the result of decision processes, which decide for the best product (for the actor at that time), based on which the actor adopts this product in order to fulfill a need.

0.23 A flexible option for a process model is the use of an **event scheduler** for the execution of dynamic aspects of the model. In this design strategy, dynamics are represented through events affecting entities at a specific point in time, encapsulating temporal dynamics. Functional differentiation of events can be found for **communication events** including advertisement and consumer messages within their social network, **need events** as a motivation for triggering an adoption decision process, **post-purchase evaluation events** for product quality assessment[2], **market introduction events** and **product discontinuation events** for changing the status of products.

0.24 In addition to temporality, spatiality is an important characteristic in a range of different eABMs (Rai & Robinson 2015; Sopha et al. 2013; Swinerd & McNaught 2014; Schwarz & Ernst 2009). It allows to explicitly integrate geographic location and needs to address how geometry and the positioning of entities towards one another is operationalized. Due to the different requirements, an appropriate level of spatial representation needs to be reflected. Spatial aspects of the simulation need to be captured through a **spatial model**, ideally comprising a topographic scheme (relationship with spatial environment) and a metric scheme (spatial relationship between model entities).

0.25 Providing the infrastructure for model dynamics, the **social network model** can be seen as an interconnecting system. A social network can stand for a number of social ties, sometimes even several ones within one model as described in Kostadinov et al. (2014), where friendship and trading relationships are incorporated. The social network describes the connection between actors and is a crucial aspect of ID models. Frequently the network is formalized as a directed or undirected graph with model entities for nodes and communication channels for edges. The network structure can be given explicitly such as scale-free (Delre et al. 2010), purely local (McCoy & Lyons 2014) or small-world (Sopha et al. 2013) network topology. In addition to the general structure of the social network, eABMs should include network dynamics, e.g. through dynamic edge weights and topology mutability, i.e. the change of connections within the model.

Framework Modeling

0.26 The following section presents the modeling approach and the formal definition of model components based on the requirements described in 0.6. It is structured through describing the approach, the entities (actors and products), dynamics (perception, preferences, decisions, needs, communication, and event modeling) and foundations (temporal model, process model, spatial model, and the social network).

Modeling Approach

0.27 The modeling process follows a two-tiered abstraction process for agent and product heterogeneity, formally defining the necessary structures identified in the requirement analysis. Major non-functional requirements are modularity and flexibility. In the following, this will be achieved by aggregating possible entities or mechanics into sets, and defining entities by tuples of members of these sets. These sets will often be called a scheme, of which a concrete specification appears as element in the tuple defining a component.

0.28 Similarly, for similar entities with common characteristics, a structure interpreted as the group will be used, with the entities associated with them (as derived instances) through an association function. This is illustrated by figure 2.

[2]based on the assumption that using a product gives a consumer actor access to assess the true product attribute values to some extent.

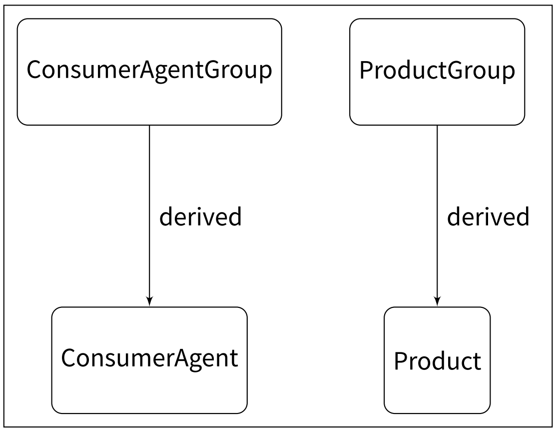

Figure 2: Two-tiered heterogeneity abstraction approach in IRPact

Actor Modeling

0.29 Actors (within this system synonymously referred to as agents) represent cognitive entities in the modeled context. These can be units of adoption (in the case of Consumer agents), shops or distribution departments (in the case of POS agents), companies or company employees responsible for sales/distribution policies of producers (in the case of Company agents) or actors within the policy sphere (as in the case of Policy agents).

0.30 Since actors derive a large part of their semantics from their interaction with dynamic components, in sake of simplicity, actors are described through parameters related to aspects that are discussed in-depth in the subsections describing the respective components.

0.31 As actors in the model are intended to exhibit cognitive processes, all actors are understood as *InformationAgents*, allowing them to provide information. Crucial for information and its interpretation in a social context is the credibility of the information source, modeled as informational authority ia, quantifying their informational credibility through a numerical value[3]. This value is used in processes involving information I.

Consumer Agents

0.32 Consumer agents describe cognitive entities that represent households or individual consumers, and are thus often the most important type of actor for the diffusion and adoption of technology. Their primary role in the model is to adopt products and to interact with other agents. As shown in figure 2, consumer agents are organized in groups, bundling common characteristics of the consumer agents and implementing consumer agent heterogeneity, elaborated on below.

0.33 A consumer agent $c \in C$ (with C being the set of consumer agents) is formalized as a 8-tupel, whose state at time t is described as
$c_t = (cag_c, ca_{c,t}, loc_{c,t}, Pr_{c,t}, Paw_{c,t}, PAP_{c,t}, ap_{c,t}, csgam_c)$, with

0.34 The Consumer Agent Group Association $cag_c = \mathfrak{cagm}(c)$ indicating which consumer agent group c is a member of (with $\mathfrak{cagm} : C \rightarrow CAG$ being the consumer agent group mapping, associating each consumer agent $c \in C$ with the consumer agent group $cag \in CAG$ it is derived from),

0.35 The state of the consumer attributes $ca_{c,t}$ of c at time t, with the vector of consumer attributes being drawn from the respective distribution CGA_{cag} of consumer agent group cag_c, in order to express consumer agent heterogeneity,

0.36 $loc_{c,t}$ representing the coordinates of c within the spatial model $\mathfrak{S}$, according to the spatial distribution $\mathscr{SD}_{cag_c}$ of the corresponding agent group cag_c,

[3] Although a universal credibility of an agent's information is unrealistic, and agent-inherent factors, their context, the nature of the information and their informational history are relevant for assessing the credibility of another agent, this would make the model a lot more complex. Since there is already little justification for informational aspects of product adoption from the analyzed innovation diffusion models, a simple approach was chosen in IRPact. More complicated informational mechanics can easily be extended or implemented using other model mechanisms.

0.37 Preference vector $Pr_{c,t}$, describing the agents' preferences $pr_v^{c,t}$ for value v at time t as entries ($Pr_{c,t} = (pr_v^{c,t})_{v \in V} \in \mathbb{R}^{|V|}$) with $pr_v^{c,t} = \mathfrak{pm}_c(v,t)$, with $\mathfrak{pm}_c : V \times T \rightarrow \mathbb{R}$ being the mapping of values $v \in V$ and time points $t \in T$ to the numerical values of the preferences of agent c,

0.38 Product awareness vector $Paw_{c,t}$ for time point t and actor c, indicating what products $p \in P$ actor c is aware of ($Paw_{c,t} = (paw_p^{c,t})_{p \in P} \in \mathbb{B}^{|P|}$, with $paw_p^{c,t}$ indicating whether c is aware of p at time t),

0.39 Product attribute perception vector $PAP_{c,t}$, indicating which perception c has of the value of the product attributes $pa \in PA$ at time t as a numerical value ($PAP_{c,t} = (papm_{pa_t}^{c,t})_{pa_t \in PA_t} \in \mathbb{R}_{\geq 0}^{|PA_t|}$, with the entries $papm_{pa}^{c,t} \in \mathbb{R}_{\geq 0}^{|PA_t|}$ as the perception c has of pa at time t, with PA_t being the set of product attributes at time t, as defined through the perceived product attribute value map $\mathfrak{ppavm}$),

0.40 The set of adopted products $ap_{c,t} = \mathfrak{pam}(c,t)$, indicating which products are adopted by $c \in C$ at time $t \in T$: $\mathfrak{pam} : C \times T \rightarrow P_{c,t}^{ad} \subset \mathscr{P}(P)$, with $\mathfrak{pam}$ being the product adoption mapping, assigning every consumer agent $c \in C$ and time $t \in T$ the products adopted by the respective consumers,

0.41 The consumer agent social graph association $csgam_c = \mathfrak{csgam}(c)$, describing which node r in the Social Graph corresponds to c, with $\mathfrak{csgam} : C \rightarrow AN$ as the consumer social graph association mapping, and AN being the set of nodes in the social graph.

ConsumerAgentGroups

0.42 To achieve the flexibility and balance between homogeneity and heterogeneity discussed in 0.8, IRPact allows for flexible consumer agent groups ($cag \in CAG$) who describe types of consumer actors through abstractions of the consumers. In addition to parameters linked to model dynamics, a cag specifies the distribution concrete values of its members are based upon. ConsumerAgentGroups thus serve as a template or blue print for consumer agents that play a decisive role in consumer agent instantiation, where its (numerical) values are drawn from the corresponding distribution, allowing for fine control over the homogeneity and heterogeneity of agents grouped together. A cag is formalized as an 11-tupel:
$cag = (CGA_{cag}, \mathscr{SD}_{cag}, \mathscr{CPD}_{cag}, \mathscr{PAWD}_{cag}, \mathfrak{ppsm}_{cag}, d, NDS_{cag}, CS_{cag}, \mathscr{IFPAD}_{cag}, \mathscr{FPAD}_{cag}, ia_{cag})$
with

0.43 The consumer group attribute vector CGA_{cag} containing the distributions the corresponding consumer agent attributes ca_c of members of the agent group are drawn from,

0.44 The spatial distribution $\mathscr{SD}_{cag}$ the coordinates of agents of cag within the spatial model $\mathfrak{S}$ are based on (i.e. $loc_{c,0}$ drawn from),

0.45 The preference distribution $\mathscr{CPD}_{cag}$ the initial preferences of the groups' agents are drawn from,

0.46 The set of product awareness distributions $\mathscr{PAWD}_{cag}$ associated with cag, the initial awareness of its agents are based on,

0.47 The product perception scheme mapping $\mathfrak{ppsm}_{cag} : PGA \rightarrow PS$, assigning every product group attribute $pga \in PGA$ its perception scheme $ps \in PS$, governing how the perceptions $PAP_{c,t}$ for derived consumers $c \in C$ are governed,

0.48 The decision process d that agents of cag employ,

0.49 NDS_{cag} being the need development scheme that agents of cag follow,

0.50 The communication scheme CS_{cag} of the ConsumerAgentGroup cag, specifying how messages and the corresponding communication events are created,

0.51 The set of initial fixed product adoption distribution $\mathscr{IFPAD}_{cag}$ of cag for the respective fixed products fp (describing how the respective product is disseminated within cag at time $t = 0$),

0.52 The fixed product awareness distribution set $\mathscr{FPAD}_{cag}$ of cag for the respective products fp, detailing how product awareness instantiated for fixed products at $t = 0$,

0.53 ia_{cag} being the information authority agents c of this ConsumerAgentGroup have.

POS Agents

0.54 As derived in the requirement analysis, a point of sale (POS) is parameterized through their dynamic behavior, location in space (if applicable) and a portfolio of available products with a POS-specific prize, allowing to investigate supply limitations and roll-out strategies.

0.55 Formally, a point-of-sale $pos \in POS$ (with POS being the set of points-of-sale within the model) at simulation time t is described as a 5-tupel
$pos_t = (Av_{pos}(t), PPr_{pos}(t), loc_{pos}, ia_{pos}, PuPS_{pos})$ with

0.56 Their product availability vector $Av_{pos}(t) = (av_p^{pos,t})_{p \in P} \in \mathbb{B}^{|P|}$, indicating whether the respective products p are available at pos at the time t,

0.57 The POS price vector $PPr_{pos}(t) = (PPr_p^{pos,t})_{p \in P} \in \mathbb{R}_{\geq 0}^{|P|}$ at time t, relative to a reference price,

0.58 loc_{pos} being its placement in the spatial model $\mathfrak{S}$ (as coordinates), analogous to consumer agents,

0.59 ia_{pos} being the information authority of the pos as an informational agent,

0.60 $PuPS_{pos}$ as the purchase process scheme modeling the purchase process at this POS.

Company Agents

0.61 In compliance with the requirements, a company agent $coa \in COA$ makes management decisions, carries out product management (as manipulation of supply) and the market introduction of products and discontinuation of products and executes advertisement and marketing (as manipulation of product perceptions). However, they do not assume a physical position, are not part of the social network and only send unidirectional messages. Their state at time t is described as:

0.62 $coa(t) = (PP_{coa}(t), PQMS_{coa}, MDS_{coa}, AS_{coa}, ia_{coa})$, with

0.63 $PP_{coa}(t) \subset P_t$ being the product portfolio of agent coa at time t, describing what products the company agent manages,

0.64 $PQMS_{coa}$ is the product quality manipulation scheme, representing how the product attributes of the products in the product portfolio of the coa can be manipulated,

0.65 MDS_{coa} standing for the management decision scheme coa uses, describing how the states of products in the product portfolio are managed and how these decisions are taken by the agent,

0.66 their advertisement scheme AS_{coa} formalizing how advertisement messages for products in their portfolio $PP_{coa}(t)$ are sent,

0.67 and ia_{coa} as their information authority.

Policy Agent

0.68 The policy agent is modeled by a singular agent who manages existing policies and introduces new ones, bundling regulatory aspects falling in the public sphere. Policies fall into three kinds of categories, namely product-directed policies, prohibitive policies and consumer-directed policies, which are formalized through schemes. The policy agent poa is formalized as follows:

0.69 $poa = (PPS_{poa}, CPS_{poa}, RPS_{poa}, MES_{poa}, ia_{poa})$, with

0.70 PPS_{poa} being the product-directed policy scheme, describing when product attributes of what products are manipulated,

0.71 the CPS_{poa}, as the consumer-directed policy scheme, specifying what policies are used by the policy agent to influence perceptions and preferences of consumer agents,

0.72 RPS_{poa} as the regulatory policy scheme, formalizing restrictions through discontinuation of products,

0.73 MES_{poa}, the market evaluation scheme, describing how the policy agent derives information about the market and its actors,

0.74 ia_{poa} as the informational authority of the policy agent pa as an information agent.

Products

0.75 Products are modeled as the entities of adoption. They contain a set of product attributes, describing various qualities of products, are organized in ProductGroups through the Product Group Association Map: $\mathfrak{pgam} : P \to PG$, and are further described by their status and longevity within the simulation. Just as AgentGroups, product groups serve to bundle products with similar properties. Products can be parameterized either through stochastic product initialization (meaning that product attributes are assigned using a probability distribution) or can be configured as fixed products with set values.

0.76 A product $p \in P$ in the set of products P is defined as a 3-tupel $p(t) = (PA_p(t), pas_{p,t}, \mathscr{PLD}_p)$ with

0.77 Its set of product attributes $PA_p(t)$ at time t with entries $pa_t = (pav_{pa,t}, pam_{pa}, pao_{pa}) \in PA_p(t)$,

0.78 Its product activation status $pas_{p,t} = \mathfrak{apm}(p,t)$ (as defined through the adopted product map, indicating whether the product is already introduced and not yet discontinued in the market at a given time t),

0.79 $\mathscr{PLD}_p$ being the product lifetime distribution of the product, specifying how long a product can be used upon adoption before it has to be readopted.

Product Attributes

0.80 Product attributes $pa_t \in PA$ describe the qualities of products quantitatively, that is holding numerical values on a number of quality dimensions (one for each product attribute), and specifying whether they are mutable and how well they can be observed by the actor. Product attribute values on these quality dimensions are described through the product attribute value map $\mathfrak{pavm}(pa,t)$ at simulation time t.

0.81 In addition to the objective value of the product attribute $pav_{pa,t} = \mathfrak{pavm}(pa,t)$, a product attribute is described by the mutability $pam_{pa} = \mathfrak{pamm}(pa) \in \mathbb{B}$ defined by the product attribute mutability map $\mathfrak{pamm}$ and observability $pao_{pa} = \mathfrak{paom}(pa) \in [0,1]$ given by the product attribute observability map $\mathfrak{paom}$. Product attribute mutability describes whether the values of a product attribute are allowed to change over the course of the simulation, as they are manipulated by certain model mechanics, whereas observability describes to what extent its true quality can be assessed by an actor.

0.82 A product attribute $pa_t \in PA$ can thus be described as a 3-tupel $pa(t) = (pav_{pa,t}, pam_{pa}, pao_{pa})$ over time.

Product Groups

0.83 In analogy to Consumer Agent Groups, product groups serve to bundle products with similar properties and to balance homogeneity and heterogeneity through the use of probability distributions for the description of their attributes, and to specify relations between these.

0.84 A product group $pg \in PG$ is formalized as an 8-tupel $pg = (PGA_{pg}, PPG_{pg}, EPG_{pg}, FP_{pg}, pgn_{pg}, SP_{pg}, odp, \mathscr{PLD}_{pg})$, with

0.85 The set of product group attributes PGA_{pg} associated with pg,

0.86 The set of prerequisite product groups PPG_{pg} for the product group,

0.87 The excluding product groups EPG_{pg},

0.88 FP_{pg} as the set of fixed products of product group pg,

0.89 The product group need $pgn_{pg} = \mathfrak{pgnm}(pg)$ a product group fulfills (with the product group need map $\mathfrak{pgnm} : PG \rightarrow \mathscr{P}(N)$),

0.90 The standard product SP_{pg} for the product group pg,

0.91 The overwrite decision process $odp = \mathfrak{odpm}(pg)$,

0.92 and $\mathscr{PLD}_{pg}$ being the (default) product lifetime distribution used for products derived from this product group.

ProductGroupAttributes

0.93 Formalizing the qualities of product groups, product group attributes are the most fundamental aspect of product groups. Understood as qualities of products, every product is assigned a scalar value on the quality dimension, its mutability and observability. Since product groups serve as blue prints for products, instead of taking on concrete values on their product attribute values, product groups describe these through probability distributions, while mutability and observability are constant over all ProductAttributes derived from this ProductGroupAttribute.

0.94 Analogous to product attributes, a product group attribute is formalized as a 3-tupel $pga = (\mathscr{PGAVD}_{pga}, pgam_{pga}, pgao_{pga})$ with

0.95 $\mathscr{PGAVD}_{pga}$ being the distribution the product attribute values are drawn from via realizations of the random variable $X_{PGAVD_{pga}}$,

0.96 $pgam_{pga} \in \mathbb{B}$ being the product attribute mutability,

0.97 $pgao_{pga} \in [0, 1]$ being the product attribute observability.

0.98 The mapping between the product group attributes and the attributes belonging to them is done through the product group attribute product attribute mapping $\mathfrak{pgapam} : PA_p \to PGA$, with $\mathfrak{pgam}(p) = pg$. With the exception of fixed products, the respective initial value of the product attribute $pav_{pa,0}$ is determined through $X_{PGAVD_{pga}}$ with $pga = \mathfrak{pgapam}(pa)$.

Fixed Products

0.99 In contrast to assigning attribute values through a stochastic process, fixed products $fp \in FP = \bigcup_{t \in T} FP_t$ exhibit predetermined initial values $pav_{pa,0}$ for the respective $PA_{fp}(0)$. Fixed products are generally used to parameterize scripted events, that is the market introduction or discontinuation of a product.

0.100 The initial relation of consumer agents and existing fixed products is described through the respective fixed product awareness distributions $\mathscr{FPAD}_{cag}^{fp}$ (specifying how awareness about fp is initially distributed in cag) and the initial fixed product adoption distributions $\mathscr{IFPAD}_{cag}^{fp}$ (describing how the adopters of fp at the beginning of the simulation are distributed). Fixed products themselves are formalized just like products, and only bypass the stochastical generation step:

0.101 $fp(t) = (PA_{fp}(t), pga_{fp}, pas_{fp,t}, \mathscr{PLD}_{fp})$,

0.102 The product lifetime distribution $\mathscr{PLD}_{fp}$ can differ from other products in the product group, but will often correspond to the one used for stochastically initiated products.

ProductGroup Relations

0.103 Restrictions on the adoption of products are modeled through the prerequisite PPG_{pg} and excluding product relations EPG_{pg} of the respective product groups. These are sets of other product groups out of which a consumer agent needs to have adopted a product before it is possible for them to adopt the specified product group (in the case of PrerequisiteProductGroups), or which exclude the adoption of said product (in the case of the ExcludingProductGroup).

Decision Overwrite

0.104 Although usually the decision process that agents use is situated with the corresponding agent group, some models might have a need for 'overwriting' the decision process, which is modeled by the decision overwrite property of product groups (as $odp_{pg} = \mathfrak{odpm}(pg)$, if defined). If a decision overwrite property is set, agents use the decision process specified instead of the decision process associated with their agent group, unless competing or ambiguous overwrites exist.

Perception Modeling

0.105 Perception is modeled in a two-tiered fashion: perceptions of the values of a product attribute pa for consumer agent c at time t are described through the perceived product attribute value map $\mathfrak{ppavm} : C \times PA \times T \to \mathbb{R}_{\geq 0}$.[4] Their temporal dynamics abstract some of the technicalities of the $\mathfrak{ppavm}$ away through perception schemes $ps \in PS$ describing the initial status of the $\mathfrak{ppavm}$, e.g. as the perceived product group attribute value distributions $\mathscr{PPAVD}_{cag}$ associated with consumer agent group cag[5], and the temporal dynamics (i.e. $(\mathfrak{ppavm}(c, pa, t) | \mathfrak{ppavm}(c, pa, \hat{t}), \hat{t} < t)$).

0.106 The association of the product group attributes and the respective perception schemes is formalized through the product perception scheme mapping $\mathfrak{ppsm}_{cag} : PGA \to PS$.

[4] Which is a partial function, since it is not defined for values of products a consumer is not aware of (that is $\mathfrak{ppavm}(c, pa, t) = \text{undef} \Leftrightarrow \mathfrak{pavm}(c, p, t) = \text{false}, p : pa \in PA_p$).

[5] As $\mathscr{PPAVD}_{cag}$ depends on the perception scheme associated with cag, it only indirectly defines consumer agent groups and is not part of their constituting tuple.

Product Awareness

0.107 While product attribute perception captures the imperfect information consumers have of the true value of a product attribute, product awareness describes whether a consumer is aware of a product (if a product is on the market). Knowledge of a product's existence is modeled through the product awareness map $\mathfrak{pawm} : C \times P \times T \to \mathbb{B}$, describing whether c is aware of p at time t ($\mathfrak{pawm}(c, p, t) = \text{true}$). It is initially parameterized through the product awareness distributions $\mathscr{PAWD}^{pg}_{cag}$, and has the value *true* after product encounter, e.g. through communication about the product by another consumer agent, company originated consumer agent messages or by encountering a product in a POS.

Preference Modeling

0.108 Moral or ethical dimensions of consumer actors are modeled through *values* $v \in V$ and the importance (strength) actors assign to them, as a numerical value. Preferences are formalized through the preference map $\mathfrak{pm}_c : V \times T \to \mathbb{R}_{\geq 0}$ assigning a numerical value to the strength of value $v \in V$ for consumer c at simulation time t. The relation between preferences and product attributes is modeled as a weighted map, the (temporally static[6]) *product attribute value preference mapping* $\mathfrak{pavpm} : PA \times V \to \mathbb{R}_{\geq 0}$, in which each product attribute is associated with one or several values in order to relate customer preferences (quantified values), to be used in product evaluation in the purchase decision process.

Adoption Decision Modeling

0.109 At the core, simulating product adoption is about modeling the decisions of consumer agents between suitable products. Products are no end in itself, however, and are employed to satisfy the *needs* of consumers.

0.110 In IRPact, this is formalized through decision processes d associated with a consumers' agent group, representing the cognitive processes they employ for deciding between suitable products. From the perspective of the model, *needs* the products' *ProductGroup* fulfills are satisfied with the adoption.

Decision Process Modeling

0.111 Formally, decision processes are specified through the product adoption decision map $\mathfrak{padm} : C \times \mathscr{P}(P) \times T \times D \to P$, $\mathfrak{padm}(c, P_{pot}, t, d) = p$, describing the taken product adoption for product p of actor c with the eligible potential products P_{pot} at time t for decision process d. For decision processes in which the state of actors in the social network of an agent or other system aspects are used, this is extended to the extended product adoption decision map: $\mathfrak{epadm} : \mathfrak{M} \times T \to P$, with $\mathfrak{M}$ representing the model as a tuple of all its components.

0.112 Usually the decision process is triggered when a need event is processed and is thus governed by the process model, often depending on the internal state of the agents and certain environmental parameters; However, decision processes can be executed at another time depending on the modeled system dynamics, governed by the process model.

0.113 When a product p is adopted (i.e. $\mathfrak{padm}(c, P_{pot}, t, d) = p$), it is added to the set of adopted products $ap_{c,t} = \mathfrak{pam}_{c,t}$ of actor c for the lifetime of the product adoption, as drawn from the respective distribution $\mathscr{PLD}_{pga}$ at adoption time. This is formalized through the partial product attribute lifetime map $\mathfrak{palm} : C \times T \times P \to \mathbb{B}$, describing whether an adopted product is operational (i.e. in a state to satisfy the associated need) at a given time, and is related to $\mathfrak{pam}(c, t)$ by the following:

0.114

$$\mathfrak{palm}(c, t, p) = \begin{cases} 1 & p \in \mathfrak{pam}(c, t) \wedge t \in [\hat{t}, \hat{t} + x^{pl}_{p,c,\hat{t}}] \\ 0 & \text{else} \end{cases}$$

with $\hat{t}$ being the time the product is adopted[7], i.e. $\hat{t} = \arg\min_t p \in \mathfrak{pam}(c, t)$, and $x^{pl}_{p,c,\hat{t}}$ being the realization of the random variable $X^{c,t}_{pl}$ for product p adopted by consumer c at time $\hat{t}$.

[6] with the potential exception of the market introduction of new products, where a new mapping is added.

[7] To not make it overly complicated, although technically wrong, the authors chose to express it like this, although naturally the same product could be adopted several times after it expired; This is respected in the implementation, but omitted here for clarity.

Needs Modeling

0.115 Needs $n \in N$ derive their semantics from products satisfying them, decision processes, and the process describing how they arise. The latter is called a *need development scheme* NDS_{cag}, associated with *ConsumerAgentGroup* cag, which is invoked by the process model to create an ordered list of needs. Need development schemes differ greatly by their dynamics, and are specified by what happens with products if their lifetime is exceeded, when previously satisfied needs are not fulfilled and how need satisfaction is mediated with the replacement of adopted products when products are discontinued.

0.116 Needs are associated with a *needIndicator* for every *ConsumerAgentGroup* through the need indicator function $\mathfrak{ni} : CAG \times N \to \mathbb{R}_{\geq 0}$, associating a numerical value $\mathfrak{ni}(cag, n)$ with need n and ConsumerAgentGroup cag, within the NDS_{cag}, specifying the needs interpretation, but gaining their semantics from other components.

Social Network Modeling

0.117 In order to achieve the largest generality, the social network is modeled as a dynamic directed weighted (multi-) graph with actors as nodes and connections between actors as edges. For models interested in the network dynamics, IRPact allows for temporal changes in the social network. Central to the social network is the structure of communication channels between consumer agents through different media (such as communication flow, information flow, friend-of-relations etc.). Further dynamic aspects implemented in IRPact are change of edge weights and network topology.

0.118 The social network is modeled as a 4-tupel $SN = (G, \mathfrak{w}, ews, tms)$ with the social graph $G = (AN, E)$, the edge weight function $\mathfrak{w}$ and its manipulation scheme ews, and the topology manipulation scheme tms. At the core of the social network is the social graph $G = (AN, E)$, describing the relationships between agents as nodes $r, o \in AN$ through directed edges $e = (r, o, m) \in E$ (from node r to node o), where each edge is associated with a medium $m \in M$. A node r is associated with a consumer agent c through the consumer agent social graph association mapping $r = \mathfrak{csgam}(c)$.

0.119 The edges are further associated with different interactions through the medium[8], as specified in other model components representing different qualities of information flow.

Edge Weight Schemes

0.120 Edges in the social graph are associated with a weight through the edge weight function $\mathfrak{w} : E \times T \to \mathbb{R}_{\geq 0}$, allowing for dynamic behaviour and heterogeneous edge weights. In order to balance flexibility and manageability of the function, edge weight schemes $EWS \ni ews = (ewis, ewds)$ are used. These describe under which dynamics the edge weights are determined / calculated in the course of the simulation, and include the edge weight initialization scheme $ewis \in EWIS$ (specifying $\mathfrak{w}(e, 0)$) and edge weight dynamics scheme $ewds \in EWDS$ (for $\mathfrak{w}(e, \hat{t}), \hat{t} > 0$).

Topology Mutability Schemes

0.121 The topology mutability scheme $tms \in TMS$ abstracts the change of the topology of the social network, using the topology manipulation function $\mathfrak{tmf} : AN \times AN \times T \to \mathbb{B}$, indicating whether at time $t \in T$ an edge exists between node $r \in AN$ and node $o \in AN$, allowing for new edges to be added to the social network, as well as for having them removed. With this, the edge set depends on the simulation time and the social graph is described as $G = (AN, E) = (AN, E_t)$.

0.122 Another aspect of the topology manipulation is the possibility of self-reference within the social graph, meaning edges from one node to themselves ($(r, r, m) \in E$ for $r \in AN, m \in M$). Depending on the semantics of the edge (or edge type), this is further specified by the topology manipulation scheme.

Communication Modeling

0.123 In IRPact, perceptions are shaped primarily through reported experiences of communicating adopters and word-of-mouth of other consumers. Consumers exchange product perception manipulation messages $CPOM \ni$

[8] Such as information exchange, value exchange, trust, marketing information etc.

$copm = (c, \hat{c}, pa, p)$ based on their own perception, which change the (perceived) product attribute values of product attribute pa of product p of the receiving agent $\hat{c}$, depending on the perceived product values of the sender c of the message. In addition to the messages sent by consumer agents, company agents can also send (product perception manipulation) messages as specified through the advertisement scheme.

0.124 Upon receiving a product perception manipulation message, the receiver becomes aware of it if they aren't already and the product p is active within the simulation, and a new perception of pa is added, based on the perception $\mathfrak{ppavm}(c, pa, t)$ of the sender c and the weight of the edge connecting them in the social graph (i.e. $\mathfrak{w}((r, o, m), t)$, with $r = \mathfrak{csgam}(c), o = \mathfrak{csgam}(\hat{c})$). How this perception is incorporated into the cognitive context of the receiving agent $\hat{c}$ depends on their perception scheme (as parameterized through the $\mathfrak{ppsm}_{cag}$, with cag being the respective consumer agent group of $\hat{c}$).

0.125 Message scheduling is governed by the CommunicationScheme CS_{cag}, determining when communication events are sent through the social network. Formally, this is described by the communication event mapping $\mathfrak{cem}_{cag} : COPM_{cag} \times T \to CE_{cag}$, formalized by the *ConsumerAgentMessageScheme*. A crucial component of these are the message activity distributions $\mathscr{MAD}_{cag}$, which characterize the number of messages per time unit the groups consumer agents send to connected consumer agents $\hat{c}$.

0.126 Messages creation is governed by message schemes, depending on the sender of the message and the state of the simulation, formalized as $\mathfrak{ms}_{cag} : C \times \mathfrak{M} \times T \to \mathscr{P}(COPM)$, describing which messages consumer agents $c \in C$ with $\mathfrak{cagm}(c) = cag$ send at time $t \in T$ depending on the model state $\mathfrak{M}$. Together, these schemes form the communication scheme $CS_{cag} = (\mathfrak{ms}_{cag}, \mathfrak{cem}_{cag})$.

Information and Advertisement Modeling

0.127 Information seeking is governed by schemes invoked by the process model or the respective decision processes, describing what information actors think they require and what strategies they use to find it. Information in IRPact is understood in a naive form, as a piece of knowledge $i \in I$, with I the set of all information, originating from an information agent. Forms of handling information are specified through information schemes ($is \in IS$ as pair $is = (\mathfrak{iavm}, \mathfrak{pim})$), which specify how information model mechanics work and and thus describe the information ecosystem within the simulation through their availability and product information seeking behavior.

0.128 Information is processed by an agent according to the perception scheme by acting as a perception respective to the product attribute (specified by the information product attribute mapping function $\mathfrak{ipam} : I \to PA$, defined for all product attribute information $pai \in PAI \subset I$).

0.129 Information schemes IS specify what information is available for agents to process via the information availability map $\mathfrak{iavm} : I \times T \to \mathbb{B}$, indicating whether a given information is 'known' at a given time $t \in T$, and how product information is sought by consumer agents through the product information map $\mathfrak{pim} : C \times PA \times T \to \mathscr{P}(I)$, describing what information about a product attribute $pa \in PA$ are relevant to a consumer agent $c \in C$ at time $t \in T$. These are used as a basis for deriving relevant information events.

0.130 Advertisement stems from company agents, through their *advertisement scheme* AS_{coa}, generating *product perception manipulation messages*, as detailed above.

Event Modeling

0.131 A number of processes in IRPact are triggered by events, which bind semantics to the temporal evaluation of their desired effect. Different events are detailed in the following.

Scripted Events

0.132 Scripted events, as exogenously supplied events, comprise the market introduction and discontinuation of products, configured through the parameterization of the simulation and formalized by the scripted events scheduling function $\mathfrak{sesf} : SE \to T$. Scripted events are associated with fixed products fp that are introduced or discontinued, formalized through the scripted product event association mapping $\mathfrak{speam} : SE \supset SPE \to FP$, with SPE being the scripted product events as subset of the scripted events.

0.133 While market introduction events *activate* products (allowing them for adoption, i.e. $\mathfrak{apm}(p, t) =$ true $\forall t \geq \hat{t}$), product discontinuation events make a fixed product unavailable to customers (i.e. $\mathfrak{apm}(p, t) =$ false $\forall t \geq \hat{t}$) for an event at scheduled time $\hat{t}$.

Need Events

0.134 Products that a consumer is aware of, that are available to the market, that are associated with the need, that are not excluded by single adoption and that are not excluded by products already adopted, enter a decision process, which is either a commonly overwritten decision process odp of the products or the decision process associated with the actor. This is mediated by need events, as described through the *NeedDevelopmentScheme*.

Post-Purchase Evaluation Events

0.135 Post-purchase evaluation adds a product perception to the consumer's product attribute perception with a weight based on the observability of the product and the true product attribute value:
$\mathfrak{ppavm}(c, pa, t)|(\mathfrak{ppavm}(c, pa, \hat{t}), poa_{pa}, pav_{pa,\hat{t}})$

Information Events

0.136 Information events link information $i \in I$, the consumer agent $c \in C$ interpreting it and the time of information evaluation $t \in T$ as a triple $ie = (i, c, t)$. Their evaluation adds a perception with the strength of the information authority ia_c of the agent c the information originates from, as formalized through the information authority mapping $\mathfrak{iam} : IA \rightarrow \mathbb{R}_{\geq 0}$ information agents exhibit, with IA as the set of information agents.

Temporal Modeling

0.137 In order to govern the temporal dynamics for the model entities, the temporal model is hierarchically situated above the process model, and drives the process model within the temporal frame. The temporal model further serves as a frame of reference.

0.138 Different time models allow the system to follow different timing regimes. This comprises the discrete and continuous time model. In general, discrete time models operate in steps that advance the time of the simulation step by step (usually $T \subset \mathbb{N}_0$), and thus only model dynamics corresponding to these time points are valid within the model, whereas in continuous times regimes all time points prior to the simulation end time are valid (i.e. time is continuous), and the mechanism governing simulation time is more event-driven. This is described by $T \subset (t \in \mathbb{R}_+ | t \leq t_{end})$, with $t_{end} > 0$ the end time of the time horizon of interest.

Process Modeling

0.139 The process model specifies how actors are processed, and is based on the corresponding event scheduler, especially in the case of continuous temporal models. Another aspect specified by the process model is how actors act when ceasing to adopt a product. This is governed by the adoption replacement scheme.

0.140 The process model is specified as the pair $PM = (es, ars)$ of the event scheduler $es \in ES$ and the adoption replacement scheme $ars \in ARS$, which describes how actors act after a product has been discontinued (e.g. by creating a need event).

Spatial Modeling

0.141 To depict an appropriate level of spatial representation, the framework uses a topographic scheme and a metric scheme, with the topographic scheme TS governing the spatial relationship between actors, and the metrical scheme MS determining how terms like distance are understood.

0.142 Whereas (spatial) actors are positioned through their coordinates loc, their positioning towards the geometry of other spatial model aspects such as the model border (by the topographic scheme) is governed by the spatial border map $\mathfrak{sbm} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B}$. Where this map describes whether a coordinate loc falls within the borders of the model, the metrical scheme determines the distance between two entities (by embedding the spatial model in the respective metric space). Together, these schemes describe the spatial model $\mathfrak{S} = (TS, MS)$.

0.143 Entities within the simulation are further associated with a spatial distribution for the geographical initialization of the model. These entities are thus assigned a position in the spatial model according to the respective distribution.

Software Implementation

0.144 Based on the model design described above, a software implementation was written in Java 1.8 with publicly available source code. Build automation was done with maven, and the libraries used were *apache commons lang, apache commons math, colt, jackson, jadex, junit, log4j and sqlite jdbc*. Due to format constraints, the following only attempts to give an overview of the implementation. A javadoc-based documentation and a structured discussion can be found in the public repository. The model is archived on the CoMSES Computational Model Library.

0.145 The code base is designed to be highly object-centered, and is organized in 24 packages and 273 classes, which are based on the model description with component abstractions implemented as abstract classes. Group-instance relationships are realized via attributes and are modeled as instances-partOf relations.

0.146 Implemented classes model entities, scheme definitions, scheme instances, configuration objects, input-output, simulation behavior, factories, auxiliary data structures, class hierarchies and helper functionality.

0.147 With the exception of helper objects and objects encapsulating processes, the system is constituted by entities. This is formalized by the abstraction of a *SimulationEntity* from which all entities are derived. In order to easily refer to other system components within the entities, the system is encapsulated in a *SimulationContainer*, to which each *SimulationEntity* refers to.

Actor Modeling

0.148 Conceptually, actors within IRPact are implementations of the *Agent* class, an extension of *SimulationEntity*. Agents that are situated spatially, further extend this to the abstraction of a *SpatialAgent*, being equipped with (2-dimensional) coordinates. Additionally, agents that (potentially) dispense information are implementations of the *InformationAgent* class, which through the *informationAuthority* attribute implements the information authority ia many agents are modeled with.

0.149 (Synchronous) Consumer agents are managed through the *SynchronousConsumerAgentFactory*, which manages the creation of the agents, while Consumer Agent Groups are modeled through the *ConsumerAgentGroup* class, regardless of synchronousity.

0.150 *POSAgents* are implemented as a *SpatialInformationAgent*, a more specific child class of a *SpatialAgent*, modeling both *POSAgent* spatiality and the information authority ia_{pos}.

0.151 Company agents are managed through the *CompanyAgentFactory*, which manages the creation of the agents, their IDs and names as well as the schemes that characterize the company agents.

0.152 As with the company agent, a policy agent's behavior is mediated through schemes. The product-directed policy scheme, consumer-directed policy scheme, regulatory policy scheme and market evaluation scheme are implemented as interfaces, with the corresponding methods. Each scheme used must implement these interfaces in order to provide the respective functionality of the schemes. When and how these schemes are executed depends on the process model and needs to be specified there. The policy agent is a global, singular agent, and is managed by the *PolicyAgentFactory*.

Product Modeling

0.153 Since products cover a number of aspects and come in different (organizational) forms within the simulation, modeling products is dispersed over a number of classes: parameterization of products is encapsulated in the *ProductConfiguration* class, they are instantiated within the *SimulationContainer* using the *ProductFactory*, and are presented through the *Product* class. In accordance with the modeling approach, further classes comprise the *ProductAttribute*, *ProductGroup* (and for technical reasons also the *ProtoProductGroup*), *ProductGroupAttribute*, *FixedProductDescription* and the *AdoptedProduct* class.

Perception Modeling

0.154 Product (attribute) perception is governed by two concepts as different levels of abstraction, namely the (abstract) *ProductAttributePerceptionScheme* class and the *PerceptionSchemeConfiguration* of the respective *ProductGroupAttribute*. The perception scheme describes the actual perception on the level of ProductAttributes and ConsumerAgents, whereas the perception scheme configuration specifies what *ProductAttributePerceptionSchemes* are chosen for the respective ProductAttributes and ConsumerAgents, as well as how they are initialized, on the level of ProductGroupAttributes and ConsumerAgentGroups. By this, the *ProductAttributePerceptionScheme* corresponds to the perceived product attribute value map ($\mathfrak{ppavm}$), whereas the *PerceptionSchemeConfiguration* implement the product perception scheme mapping $\mathfrak{ppsm}_{\mathfrak{cag}}$. The latter thus describes on a more abstract level which *ProductAttributePerceptionSchemes* are chosen and how they are initialized. This includes what scheme is chosen (*associatedPerceptionScheme*), what parameters are assigned to it (*perceptionSchemeParameters*) and how this is initialized (*perceptionInitializationScheme*).

Preference Modeling

0.155 As preferences assign numerical values to values as $\mathfrak{pm}_c : V \times T \to \mathbb{R}_{\geq 0}$, a *Preference* is basically a comparable encapsulation of a Value and a numerical value, implemented as a double precision value. Since preferences are used to evaluate the utility of products for adoption decision processes, the preferences of the *ConsumerAgent* and the *ProductAttributes* need to be related to one another. This is done via the *ProductGroupAttributeValueMapping*, a structure that associates a numerical value to the coupling of the *ProductGroupAttribute* and the value the preference is based upon.

Decision Modeling

0.156 The decision process is modeled as an abstract class (*ConsumerAgentAdoptionDecisionProcess*), itself derived from the *DecisionMakingProcess*. It implements both $\mathfrak{padm}$ and $\mathfrak{epadm}$, depending on the scope of the concrete decision process, and offers two (abstract) methods to specify the adoption decision process and to compare products. Product adoption takes place through processing *NeedEvents*. The *ConsumerAgentAdoptionDecisionProcess* describes the process of how to weigh different product options against one another, and decide between them.

Needs Modeling

0.157 A *Need* is implemented as a basic data type characterized by just its *name* (as a String). Needs gain their semantics through other model components, such as the abstract *NeedDevelopmentScheme* class, which describes how needs develop and what happens when a product expires.

Social Network Modeling

0.158 The social network describes the dynamic structure of the interaction between the consumer agents within the simulation through a *SocialGraph* in direct correspondence with $G = (AN, E)$, the *EdgeWeightManipulationScheme* implementing the temporal dynamics of the edge weight function $\mathfrak{w} : E \times T \to \mathbb{R}_{\geq 0}$ and a *TopologyManipulationScheme* for the topology manipulation function $\mathfrak{tmf} : AN \times AN \times T \to \mathbb{B}$. In addition to the nodes and (type-annotated) edges involved in the graph directly, the *SocialGraph* provides a number of additional data structures and methods. It is managed by the *SNFactory*, which creates the respective graphs based on the configuration of the social network, the state of the simulation and the set of initial nodes within the graphs.

Communication Modeling

0.159 Communication is done through messages on the basis of a *CommunicationScheme* and a *MessageScheme*, with the scheduling and interpretation of the messages done by an *EventHandler* and a *ProcessModel*. A message in its general form is a data structure that manages a *sender* and *receiver* Agent, and offers the method *processMessage*, which invokes the type-sepcific message and gives rise to its dynamic behavior. Message creation is governed by their respective *MessageScheme*, describing what messages a given sending agent wants to send, and operationalizes $\mathfrak{ms}_{cag}$. They are put in context by being scheduled as communication events by the *CommunicationScheme*, linking their evaluation to the temporal model and implementing CS_{cag} of the respective customer group.

0.160 Messages are refined in messages coming from other consumer agents (*ConsumerConsumerPerceptionManipulationMessage*) and messages originating from a company agent (*CompanyConsumerPerceptionManipulationMessage*). How these messages are created, based on the sender of the message and the state of the simulation, is formalized in a *MessageScheme*, an abstract class that creates a set of *Messages* based on the *SimulationContainer* and the sendingAgent.

Information Modeling

0.161 As information is too abstract a concept to have any meaningful implementation in IRPact, *Information* was chosen to be an abstract class, with only a reference to the agent from which the information stems (the information originator). Information about product qualities is modeled through the *ProductAttributeInformation* (corresponding to PAI) which additionally contains a value and a *ProductAttribute*.

0.162 The way information enters the simulation and is processed is implemented by *InformationSchemes* which specify how information enters the simulation by an information agent, implementing $\mathfrak{iabm}$.

Advertisement Modeling

0.163 Where information describes the aspects of knowledge that is actively sought, *advertisement* describes knowledge that is actively transferred to an agent (in this case by a *CompanyAgent*). This is governed by *AdvertisementSchemes*, corresponding to AS_{coa}, and implemented as abstract classes, associated with a *CompanyAgentMessageScheme* and the (abstract) method *advertiseProductPortfolio*.

Event Modeling

0.164 An *Event* in IRPact is an abstraction of model dynamics, linking model aspects to the process and temporal model. From the implementation perspective, an *Event* is an abstract class, inheriting from *SimulationEntity*, that is comparable by the time it is scheduled for, and describes itself how it is processed. Events are classified as *ScriptedEvents*, *CommunicationEvents*, *NeedEvents*, *Post-PurchaseEvaluationEvents* and *InformationEvents*.

0.165 *ScriptedEvents* are predetermined events that introduce a product to the simulation (*MarketIntroductionEvent*, parameterized as *MarketIntroductionEventDescription*), allowing consumer agents to adopt them as well, or discontinuing an existing product (*ProductDiscontinuationEvent*, parameterized as a *ProductDiscontinuationEventDescription*), removing the product from the *SimulationContainer* and the agents' perception, as well as invoking the *removeProductFromAgents* method of the *AdoptionReplacementScheme*. *ScriptedEvents* thus essentially link a *FixedProductDescription* and the execution time of the corresponding event (*scheduledForTime*) to one another.

0.166 Similarly, a *CommunicationEvent* is an event that relates a *Message* to the temporal framework of the simulation, by 'wrapping' it into an *Event*, as explained above.

0.167 The way a *CommunicationEvent* links a *Message* to the temporal model, the *NeedEvent* links a *Need* to the temporal model by the *scheduledForTime* attribute. Processing a need first involves finding products relevant to satisfy the need from the perspective of the *ConsumerAgent*. These are products

- that the consumer is aware of
- that are within the market (activated)
- that satisfy the need
- that don't have any requirements or exclusive criteria preventing product adoption.

If by these criteria at least one product remains (and the need is still unsatisfied at the time of need event processing), these products enter a *ConsumerAgentAdoptionDecisionProcess*. If no product satisfies the criteria mentioned (or the need is already satisfied), the *NeedDevelopmentScheme* of the *ConsumerAgent* will be asked to schedule new *NeedEvents*.

A Post-purchase evaluation (PPE) event serves to correct the perception of an agent that interacts with the product (on a regular basis) towards the true qualities (i.e. *ProductAttributes*) of the product. As such, it links the *Product* and the *ConsumerAgent* that adopted the product, and upon evaluation adds a perception according to the true qualities of the products to the perceptions of the adopted agent, based on the observability of the corresponding *ProductGroupAttribute*.

As a data structure an *InformationEvent* is characterized by the reference to the *Information* linked to its evaluation and the *ConsumerAgent* processing the information, thus implements the triple (i, c, t). Its behavior is implemented by invoking the *processInformation* method of the respective agent, specifying the semantics of the information event there.

Temporal Modeling

0.168 The temporal model describes the temporal dynamics of the simulation. It governs the execution of the simulation and is superordinate to the *ProcessModel*. The *TimeModel* is implemented as an abstract class, containing the respective *processModel*) and *simulationTime* as well as an (abstract) method to start the simulation (*startSimulation*).

0.169 Two temporal modes exist: DISCRETE and CONTINUOUS (as modeled via the enumeration *TEMPORALMODE* in the *TimeModel* class). The discrete models is an implementation of the *DiscreteTimeModel* and the continuous one is implemented by the *ContinuousTimeModel*.

0.170 The configuration of the temporal model is given in the *TemporalConfiguration*, as given by the *TimeLoader*. The discrete time model models time in finite, discrete steps ($T \subset \mathbb{N}_0$), and is parameterized by the length in amount of steps of the simulation (*numberTotalSteps*), a boolean indicating whether the time regime is *synchronous* or not and the respective event scheduler (as a *DiscreteEventScheduler*). It progresses in discrete steps, executing all *Events* corresponding to the step. Consequently, the *ProcessModel* is invoked, and processes are executed based on this. The continuous time model considers each time point prior to the end point as being valid within the simulation (i.e. $T \subset (t \in \mathbb{R}_+ | t \leq t_{end})$). It is parameterized by the container of the simulation (*SimulationContainer*), the *ProcessModel* to use and the *SimulationLength* t_{end}.

Process Modeling

0.171 The (abstract) *ProcessModel* governs the execution of dynamic aspects of the model within the *TemporalModel*, as a subordinate model. In discrete time modeling, the process model specifies the order of execution of processes occurring within a time step, in continuous time models the process model is more focused on specifying the order of the execution of steps in a temporal interval (i.e. relative to one another). The process model specifies how actors are processed and how actors act when ceasing to adopt a product (through the *AdoptionReplacementScheme*). In most cases it is based on the corresponding *EventScheduler* (corresponding to es).

0.172 Events are managed by an *EventScheduler*, which is itself an abstract class that governs the order of the execution of the events and offers a number of stack-like operations for event handling. Since events are executed in a temporal reference frame, the temporal model is of crucial importance. For this, the *DiscreteEventScheduler* and the *ContinuousEventScheduler* child classes are used.

0.173 The abstract *AdoptionReplacementScheme* describes how products adopted by consumer agents are replaced once they are discontinued on the market, and how they respond to these discontinuations. This is done through the *readopt* method that describes how ConsumerAgents behave when they readopt a product (or how they choose a similar product that satisfies the needs) and the *removeProductFromAgents* method that describes what happens when a product gets 'removed' from the agents, such as causing them to readopt at some point, to seek information or act in any other way.

Spatial Modeling

0.174 The *SpatialModel* is an abstract class that is situated within a *SimulationContainer* and describes the spatial character of the simulation. It defines how the metric spatial entities are situated to one another relatively (as a string *metric*, corresponding to the metric scheme MS), and provides an ability to check whether a coordinate is within the spatial model, operationalizing the topographical scheme TS. The spatial model is initialized through the *SpatialFactory*, based on the *SpatialConfiguration*, which specifies the characteristics of the spatial model.

Concluding Remarks

0.175 The main objective of the modeling approach was to achieve modularity and flexibility, as well as clarity through an explicit description of the concepts used. The modeling approach was formal and strongly interpretative, albeit mathematically not deep. This necessitates semantic awareness of the user of the model, but keeps the model intuitive and accessible (as long as the model aspects the framework components correspond to are respected). The structure of the model description followed the requirement analysis for a versatile and robust model and was very strongly shaped by the components and their respective requirements there.

0.176 This approach led to a three-layer model, in which the layers identified are the super agent layer, the agent layer and the sub agent layer.

0.177 The super agent layer consists of all aspects overarching the agents, namely the environment. It comprises the connection between the agents (network and communication taking place within it), the process model, the event scheduler and temporal model (structuring the model dynamics), the spatial model (as the spatial context of the individual agents), the product environment (feeding the need dynamics and its satisfaction), the scripted events and the information ecosystem.

0.178 The individual agents and model aspects directly associated with them are situated in the agent layer. This comprises the consumer agents, the consumer agent groups and their affinities towards one another, the company agents, the POS agents, the policy agent, as well as the events the agents process, mitigating model dynamics.

0.179 The aspects describing the individual agents can be associated with a third layer, the sub agent layer, which describes the aspects internal to the respective agents. It is formed by the consumer agent attributes, coordinates, preferences, awareness, perception, needs, decision processes and adopted products, the consumer agent groups attributes, their respective distributions and schemes, decision process and information authority, the POS agents product availability and price vector, their location, purchase process and information authority, the company agents' product portfolio, information authority and their schemes, as well as the policy agents schemes and information authority.

0.180 By this, a comprehensive, formally consistent and highly flexible model description has been reached, allowing model formulation, development and documentation in a concrete, comprehensive and concise manner. With this, we believe that we can consolidate the big differences in design and grounding of existing research observed in the introductory remarks and contribute a more cohesive framework, as well as a frame of discussion to the discourse on eABMs for innovation diffusion.

Declarations of interest

0.181 There are no competing interests.

Acknowledgement

0.182 Simon Johanning and Fabian Scheller receive funding from the project SUSIC with the project number 1722 0710. This study is financed by the Saxon State government out of the State budget approved by the Saxon State Parliament. Fabian Scheller also receives funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement no. 713683 (COFUNDfellowsDTU).

References

Balbi, S., Giupponi, C., Perez, P. & Alberti, M. (2013). A spatial agent-based model for assessing strategies of adaptation to climate and tourism demand changes in an alpine tourism destination. *Environmental Modelling & Software*, *45*, 29–51. doi:10.1016/j.envsoft.2012.10.004
URL http://dx.doi.org/10.1016/j.envsoft.2012.10.004

Barreteau, O., Sauquet, E., Riaux, J., Gailliard, N. & Barbier, R. (2014). Agent based simulation of drought management in practice. In *Advances in social simulation*, (pp. 237–248). Springer. doi:10.1007/978-3-642-39829-2_21
URL http://dx.doi.org/10.1007/978-3-642-39829-2_21

Bell, A. R., Robinson, D. T., Malik, A. & Dewal, S. (2015). Modular abm development for improved dissemination and training. *Environmental Modelling & Software*, *73*, 189–200. doi:10.1016/j.envsoft.2015.07.016
URL http://dx.doi.org/10.1016/j.envsoft.2015.07.016

Boero, R. & Squazzoni, F. (2005). Does empirical embeddedness matter? Methodological issues on agent-based models for analytical social science. *Journal of Artificial Societies and Social Simulation*, *8*(4)

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(Suppl 3), 7280–7287. doi:10.1073/pnas.082080899
URL http://dx.doi.org/10.1073/pnas.082080899

Broekhuizen, T. L. J., Delre, S. A. & Torres, A. (2011). Simulating the Cinema Market: How Cross–Cultural Differences in Social Influence Explain Box Office Distributions. *Journal of Product Innovation Management*, *28*(2), 204–217. doi:10.1111/j.1540-5885.2011.00792.x
URL http://dx.doi.org/10.1111/j.1540-5885.2011.00792.x

Chappin, E. J. & Afman, M. R. (2013). An agent-based model of transitions in consumer lighting: Policy impacts from the E.U. phase-out of incandescents. *Environmental Innovation and Societal Transitions*, *7*, 16–36. doi:10.1016/j.eist.2012.11.005
URL http://dx.doi.org/10.1016/j.eist.2012.11.005

Delre, S. A., Jager, W., Bijmolt, T. H. A. & Janssen, M. A. (2010). Will It Spread or Not? The Effects of Social Influences and Network Topology on Innovation Diffusion. *Journal of Product Innovation Management*, *27*(2), 267–282. doi:10.1111/j.1540-5885.2010.00714.x
URL http://dx.doi.org/10.1111/j.1540-5885.2010.00714.x

Eppstein, M. J., Grover, D. K., Marshall, J. S. & Rizzo, D. M. (2011). An agent-based model to study market penetration of plug-in hybrid electric vehicles. *Energy Policy*, *39*(6), 3789–3802

Frederiks, E. R., Stenner, K. & Hobman, E. V. (2015). Household energy use: Applying behavioural economics to understand consumer decision-making and behaviour. *Renewable and Sustainable Energy Reviews*, *41*, 1385–1394. doi:10.1016/j.rser.2014.09.026
URL http://dx.doi.org/10.1016/j.rser.2014.09.026

Glaser, B. G. & Strauss, A. L. (1967). The discovery of grouded theory. *Chicago (US): Aldine*

Graebig, M., Erdmann, G. & Röder, S. (Eds.) (2014). *Assessment of residential battery systems (RBS): profitability, perceived value proposition, and potential business models*

Kiesling, E. (2011). *Planning the market introduction of new products: An agent-based simulation of innovation diffusion*. Wien: UBW Volltext am Hochschulschriftenserver der UB Wien
URL http://othes.univie.ac.at/16743/

Kiesling, E., Günther, M., Stummer, C. & Wakolbinger, L. M. (2012). Agent-based simulation of innovation diffusion: a review. *Central European Journal of Operations Research*, *20*(2), 183–230. doi:10.1007/s10100-011-0210-y
URL http://dx.doi.org/10.1007/s10100-011-0210-y

Kostadinov, F., Holm, S., Steubing, B., Thees, O. & Lemm, R. (2014). Simulation of a Swiss wood fuel and roundwood market: An explorative study in agent-based modeling. *Forest Policy and Economics*, *38*, 105–118. doi:10.1016/j.forpol.2013.08.001
URL http://dx.doi.org/10.1016/j.forpol.2013.08.001

Macal, C. M. & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of simulation*, *4*(3), 151–162. doi:10.1057/jos.2010.3
URL http://dx.doi.org/10.1057/jos.2010.3

McCoy, D. & Lyons, S. (2014). Consumer preferences and the influence of networks in electric vehicle diffusion: An agent-based microsimulation in Ireland. *Energy Research & Social Science*, *3*, 89–101. doi:10.1016/j.erss.2014.07.008
URL http://dx.doi.org/10.1016/j.erss.2014.07.008

Palmer, J., Sorda, G. & Madlener, R. (2015). Modeling the diffusion of residential photovoltaic systems in Italy: An agent-based simulation. *Technological Forecasting and Social Change*, *99*, 106–131. doi:10.1016/j.techfore.2015.06.011
URL http://dx.doi.org/10.1016/j.techfore.2015.06.011

Rai, V. & Robinson, S. A. (2015). Agent-based modeling of energy technology adoption: Empirical integration of social, behavioral, economic, and environmental factors. *Environmental Modelling & Software*, *70*, 163–177. doi:10.1016/j.envsoft.2015.04.014
URL http://dx.doi.org/10.1016/j.envsoft.2015.04.014

Rogers, E. M. (2003). *Diffusion of innovations*. New York: Free Press, 5th ed. edn.

Scheller, F., Johanning, S. & Bruckner, T. (2018a). A review of designing empirically grounded agent-based models of innovation diffusion: Development process, conceptual foundation and research agenda. *Manuscript submitted for publication (2018)*

Scheller, F., Johanning, S. & Bruckner, T. (2018b). IRPsim: A techno-socio-economic energy system model vision for business strategy assessment at municipal level. *Research Report No.02 (2018), Leipzig University, Institute for Infrastructure and Resources Management (IIRM)*
URL http://hdl.handle.net/10419/183217

Schramm, M. E., Trainor, K. J., Shanker, M. & Hu, M. Y. (2010). An agent-based diffusion model with consumer and brand agents. *Decision Support Systems*, *50*(1), 234–242. doi:10.1016/j.dss.2010.08.004
URL http://dx.doi.org/10.1016/j.dss.2010.08.004

Schwarz, N. (2007). *Umweltinnovationen und Lebensstile: Eine raumbezogene, empirisch fundierte Multi-Agenten-Simulation*, vol. 3. Metropolis-Verlag GmbH

Schwarz, N. & Ernst, A. (2009). Agent-based modeling of the diffusion of environmental innovations — An empirical approach. *Technological Forecasting and Social Change*, *76*(4), 497–511. doi:10.1016/j.techfore.2008.03.024
URL http://dx.doi.org/10.1016/j.techfore.2008.03.024

Smajgl, A. & Barreteau, O. (Eds.) (2014a). *Empirical Agent-Based Modelling - Challenges and Solutions*. New York, NY: Springer New York. doi:10.1007/978-1-4614-6134-0
URL http://dx.doi.org/10.1007/978-1-4614-6134-0

Smajgl, A. & Barreteau, O. (2014b). Series foreword. In A. Smajgl & O. Barreteau (Eds.), *Empirical Agent-Based Modelling - Challenges and Solutions*, (pp. v–viii). New York, NY: Springer New York. doi:10.1007/978-1-4614-6134-0
URL http://dx.doi.org/10.1007/978-1-4614-6134-0

Sopha, B. M., Klöckner, C. A. & Hertwich, E. G. (2013). Adoption and diffusion of heating systems in Norway: coupling agent-based modeling with empirical research. *Environmental Innovation and Societal Transitions*, *8*, 42–61. doi:10.1016/j.eist.2013.06.001
URL http://dx.doi.org/10.1016/j.eist.2013.06.001

Stummer, C., Kiesling, E., Günther, M. & Vetschera, R. (2015). Innovation diffusion of repeat purchase products in a competitive market: An agent-based simulation approach. *European Journal of Operational Research*, *245*(1), 157–167. doi:10.1016/j.ejor.2015.03.008
URL http://dx.doi.org/10.1016/j.ejor.2015.03.008

Swinerd, C. & McNaught, K. R. (2014). Simulating the diffusion of technological innovation with an integrated hybrid agent-based system dynamics model. *Journal of simulation*, *8*(3), 231–240. doi:10.1057/jos.2014.2
URL http://dx.doi.org/10.1057/jos.2014.2

van Eck, P. S., Jager, W. & Leeflang, P. S. H. (2011). Opinion leaders' role in innovation diffusion: A simulation study. *Journal of Product Innovation Management*, *28*(2), 187–203. doi:10.1111/j.1540-5885.2011.00791.x
URL http://dx.doi.org/10.1111/j.1540-5885.2011.00791.x

Windrum, P., Ciarli, T. & Birchenhall, C. (2009). Consumer heterogeneity and the development of environmentally friendly technologies. *Technological Forecasting and Social Change*, *76*(4), 533–551. doi:10.1016/j.techfore.2008.04.011
URL http://dx.doi.org/10.1016/j.techfore.2008.04.011

Wolf, I., Schröder, T., Neumann, J. & de Haan, G. (2015). Changing minds about electric cars: An empirically grounded agent-based modeling approach. *Technological Forecasting and Social Change*, *94*, 269–285. doi:10.1016/j.techfore.2014.10.010
URL http://dx.doi.org/10.1016/j.techfore.2014.10.010

Wooldridge, M. J. (1998). *Agent technology: foundations, applications, and markets*. Springer Science & Business Media. doi:10.1007/978-3-662-03678-5
URL http://dx.doi.org/10.1007/978-3-662-03678-5

Zhang, T. & Nuttall, W. J. (2011). Evaluating Government's Policies on Promoting Smart Metering Diffusion in Retail Electricity Markets via Agent–Based Simulation*. *Journal of Product Innovation Management*, *28*(2), 169–186. doi:10.1111/j.1540-5885.2011.00790.x
URL http://dx.doi.org/10.1111/j.1540-5885.2011.00790.x

Zsifkovits, M. (2015). Agent-Based Modeling for Simulating Eco-Innovation Diffusion: A Review on the case of Green Mobility. *International Journal of Science and Research (IJSR)*, *4*(8), 264–269