Model Description IRPact: Integrated Resource Planning and Interaction

Simon Johanning

July 1, 2020

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. This will in the following be achieved by aggregating possible entities or mechanics into sets, and defining entities by tupels of members of these sets. These sets will often be called a scheme, of which a concrete specification appears as element in the tupel defining a component.

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 (1).

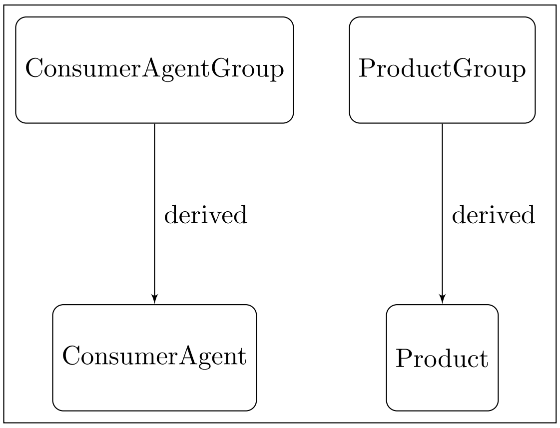

Figure 1: Two-tiered heterogeneity abstraction approach in IRPact

This discussion will exclude technical aspects related to the implementation, and instead focus on the formal description of model aspects.

1 Actor Modeling

Actors (within this system synonymously referred to as agents) represent cognitive entities in the modeled context. These can be households or individual consumers (in the case of ConsumerAgents, or units of adoption), 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).

Actors generally are intended to exhibit cognitive processes; however, due to the flexible and abstract nature of the model this is not necessary. One commonality of (practically relevant) actors within IRPact is their ability to provide information, making them *InformationAgents*. Crucial for information and its interpretation in a social context is the credibility of the information source. This is modeled through the *informational authority* as a commonality of *InformationAgents*, quantifying their informational credibility[1].

Since actors derive a large part of their semantics from how dynamic components of the system depend on them, many aspects of actors will be discussed more in-depth in the respective sections describing dynamic components, and what aspect of the semantics of actors will be modeled through these.

From the viewpoint of static components, actors can be seen as data structures which are elaborated on in the technical documentation. The different types of agents are discussed in the following.

1.1 Consumer Agents

Consumer agents describe cognitive entities representing 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 consume / adopt products and to interact with other consumer agents, and (if part of the model) also corporate and POS agents. As shown in figure (1), consumer agents are organized in groups, bundling common characteristics of the consumer agents and implementing consumer agent heterogeneity. These consumer agent groups are elaborated on below.

A consumer agent $c \in C$ (with C being the set of consumer agents) is formalized as a 9-tupel, whose state at time t is described as
$c_t = (cag_c, ca_{c,t}, loc_{c,t}, Pr_{c,t}, Paw_{c,t}, PAPM_{c,t}, d, ap_{c,t}, csgaf_c)$ with

The Consumer Agent Group Association $cag_c = \mathfrak{cagm}(c)$, indicating which consumer agent group c is a member of (with $\mathfrak{cagm} : C \to CAG$ 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; see (1.2) for more on consumer agent groups),

The state of the consumer attributes $ca_{c,t}$ of c at time t, with the vector of consumer attributes being defined by the corresponding consumer agent group cag_c, in order to express consumer agent heterogeneity,

[1]Although a universal credibility of the information of an agent is unrealistic, and both factors about the agent assessing the credibility of another agent, as well as their context, the nature of the information and their informational history are relevant for credibility assessment, 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.

$loc_{c,t}$ representing the coordinates of c within the spatial model $\mathfrak{S}$, according to the spatial distribution $\mathscr{S}\mathscr{D}_{cag_c}$ of the corresponding agent group cag_c (see (12) for more details),

Preference vector $Pr_{t,c}$, describing the agents' preferences $pr_v^{t,c}$ for value v at time t as entries ($Pr_{t,c} = (pr_v^{t,c}) \in \mathbb{R}^{\mathcal{N}_V}$) with ($\mathcal{N}_V$ being the number of values in the simulation), $pr_v^{t,c} = \mathfrak{pm}_c(v,t)$, with $\mathfrak{pm}_c : V \times T \to \mathbb{R}$ being the mapping of values $v \in V$ and time points $t \in T$ (with T being the set of valid time points in the model) to the numerical values of the preferences of agent c; for more information see (4)),

Product awareness vector $Paw_{c,t}$ for time point t and actor c, indicating what products $p \in P$ actor c is aware of ($Paw_{c,t} = (paw_p^{c,t}) \in \mathbb{B}^{\mathcal{N}_P}$, with $paw_p^{c,t}$ = true meaning that c is aware of p at time t and $paw_p^{c,t}$ = false meaning it isn't, and $\mathcal{N}_P$ being the number of products used in the simulation ($\mathcal{N}_P = card(P)$),

Product attribute perception vector $PAPM_{c,t}$, indicating which perception c has of the value of the product attributes $pa \in PA$ at time t as a numerical value ($PAPM_{c,t} \in \mathbb{R}_{\geq 0}^{\mathcal{N}_{PA}^t}$, with the entries $papm_{pa}^{c,t} \in \mathbb{R}_{\geq 0}^{\mathcal{N}_{PA}^t}$ and $\mathcal{N}_{PA}^t = card(PA_t)$ being the number of product attributes $pa \in PA$ at time t, with the entries of the vector corresponding to the product attribute pa being defined through the perceived product attribute value map $\mathfrak{ppavm}$ (see (3) for details)),

d being the decision procedure c employs for its decision processes,

The set of adopted product $ap_{c,t} = \mathfrak{pam}(c,t)$, indicating which products are adopted by $c \in C$ at time $t \in T$: $\mathfrak{pam} : C \times T \to P_{c,t}^{ad} \subset Pot(P)$, 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 a set $ap_{c,t} \in P_{c,t}^{ad}$, as the potential product combinations to adopt (subset of all products P),

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

1.2 ConsumerAgentGroups

Consumer agents are naive understood with aspects ranging from individual actor-centered aspects up to those characterizing whole groups of agents and their relationship with other groups of agents. Agents can thus be seen from an (individual) actor-centered perspective (as done in (1.1)), or from a characterization of agents by types, encompassing among others behaviors, attitudes, social and communicative patterns, perception and decision approaches. This can be formalized by assigning agents to a consumer agent group, with the intention that agents within that group are as similar as possible (intra-homogeneity) and as different as possible from other groups (inter-heterogeneity). This is often done on a socio-economic or psycho-sociological basis (such as sinus milieus, socio-economic groups, cognitive involvement in decision processes etc.), but can be done in any way the modeler desires. For this flexibility is of paramount importance, since groups should be homogeneous enough to justify group identity and sufficiently heterogeneous within a group to be 'interesting' enough.

To achieve this, IRPact allows for flexible consumer agent groups (cags, as defined through the set of consumer agent groups CAG). Cags describe types of consumer actors through abstractions of the consumers (as sketched in (1)). In addition to a number of parameters linked to model dynamics, cags feature attributes analogous to consumer actors, with the difference that instead of exhibiting concrete values for the different attributes, they are parameterized by distributions. ConsumerAgentGroups thus serve as a template or blue print for consumer agents that play a decisive role in consumer agent instantiation.

When a ConsumerAgent is instantiated, its (numerical) values are drawn from the corresponding distribution of the ConsumerAgentGroup. As an example, the preference map of agent c for value v at the beginning of the simulation is the realization of the random variable $X_{CP^v_{cag}}$ associated with the preference map distribution of the consumer agent group. Through setting / tweeking the distributions associated with aspects of the consumer agent group, the modeler can specify the degree of homogeneity and heterogeneity of agents grouped together very precisely.

A cag is formalized as an 11-tupel

$cag = (CGA_{cag}, \mathscr{SD}_{cag}, \mathscr{CP}_{cag}, \mathscr{PAW}_{cag}, \mathfrak{ppsm}_{cag}, d, NDS_{cag},$
$CS_{cag}, \mathscr{IFPC}_{cag}, \mathscr{FPAD}_{cag}, ia_{cag})$ with

The consumer group attribute vector CGA_{cag} containing the distributions the corresponding consumer agent attribute (distributions) ca_c of members of the agent group are drawn from (i.e. the entries are related 1-to-1 with each entry of the consumer group attribute vector corresponding to the consumer agent attribute derived from it and vice versa),

The spatial distribution $\mathscr{SD}_{cag}$ associated with a cag used to initialize the coordinates of agents of cag within the spatial model $\mathfrak{S}$ (and $\mathscr{SD}_{cag_c} = \mathscr{SD}_{cag}$ for $\mathfrak{cagm}(c) = cag$),

The preference distribution set $\mathscr{CP}_{cag}$ associated with cag ($\mathscr{CP}_{cag} = \bigcup_{v \in V} \mathscr{CP}^v_{cag}$, with $\mathscr{CP}^v_{cag}$ being the consumer agent preference distribution for value v for the consumer group cag; used to determine the initial preferences of an agent $c \in \mathfrak{cagm}^{-1}(cag)$),

The set of product awareness distributions $\mathscr{PAW}_{cag}$ associated with cag ($\mathscr{PAW}_{cag} = \bigcup_{pg \in PG} \mathscr{PAW}^{pg}_{cag}$, with $\mathscr{PAW}^{pg}_{cag}$ being the product awareness distribution for consumer agents belonging to cag for products belonging to product groups (see (3.1) for more information on product awareness),

The product perception scheme mapping $\mathfrak{ppsm}_{cag} : PGA \to PS$, assigning to every product group attribute $pga \in PGA$ the perception scheme $ps \in PS$ governing how the perceptions $PAPM_{c,t}$ for derived consumers $c \in C$ are gov-

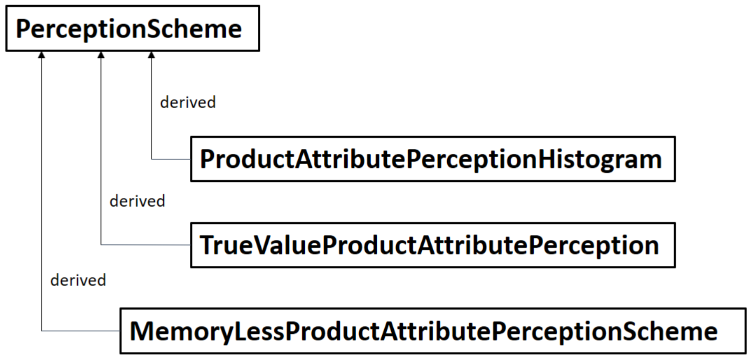

Figure 2: Relationship of perception schemes

erned (see (3) for details on modeling product perception),

The decision process d that agents of cag employ,

NDS_{cag} being the need development scheme that agents of cag follow (see (5)),

The communication scheme CS_{cag} of the ConsumerAgentGroup cag specifying how messages and the corresponding communication events are created (for more details see (8)),

The initial fixed product configuration distribution set $\mathscr{IFPC}_{cag} = \bigcup_{pg \in PG} \bigcup_{fp \in FP_{pg}} \mathscr{IFPC}^{fp}_{cag}$ of cag for the respective fixed products fp (where $\mathscr{IFPC}^{fp}_{cag}$ is the initial fixed product configuration distribution for fp within the consumer agent group cag, describing how the respective product is disseminated within cag).
$\mathscr{IFPC}_{cag}$ thus describes how the fixed product adoption is distributed for consumer agents $c : \mathfrak{cagm}(c) = cag$ of this group at time $t = 0$ (see (2.5) for more information),

The fixed product awareness distribution set $\mathscr{FPAD}_{cag} = \bigcup_{pg \in PG} \bigcup_{fp \in FP_{pg}} \mathscr{FPAD}^{fp}_{cag}$ of cag for the respective products fp. Describes how the awareness for this product is distributed for consumer agents $c : \mathfrak{cagm}(c) = cag$ of this group at $t = 0$ (see (2.5) for more information),

ia_{cag} being the information authority agents c of this ConsumerAgentGroup (i.e. $c : \mathfrak{cagm}(c) = cag$) have.

As mentioned earlier, the schemes are abstractions realized as a set, of which a concrete element is chosen for a consumer agent group. This is illustrated by the following graphic:

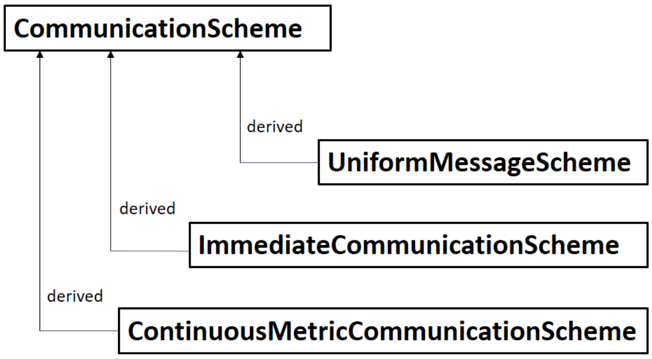

Figure 3: Relationship of communication schemes

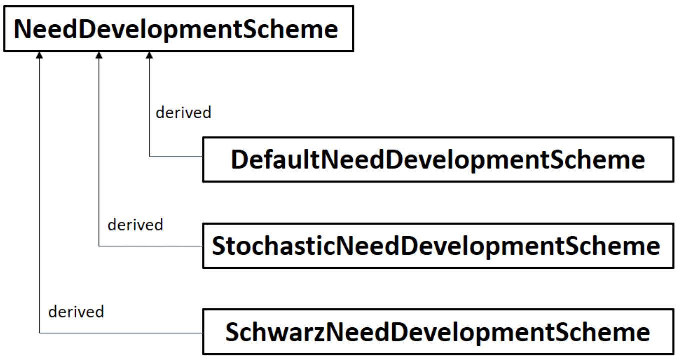

Figure 4: Relationship of need schemes

1.3 POS Agents

POS Agents represent the points-of-sales (POS) for products (or more exactly the actors 'controlling'[2] a point-of-sale). They thus represent shops or distribution / sales departments that facilitate the relation between consumers, brands and products. Depending on their nature, POS agents may be locally bound (e.g. when trading physical products) or not embedded in the spatial model (e.g. certain services or mail-order). Interaction with the POS manipulates awareness and perception of customer agents to products, and allows modeling the purchase process. With respect to their own dynamic / proactive behaviour, a POS may take decisions, employ strategies etc, set the prices for the products and manage the *availability* of the products.

A point of sale is parameterized through a portfolio of available products with a POS-specific prize, its location in space (coordinates) if it trades physical goods and is not a mail-order POS, and the dynamic behaviour bound to it. The restriction on product availability of POS' allows for evaluating supply limitations and roll-out strategies.

If the modeler is not interested in the influence of POS' for model dynamics, a single 'global' pos can be considered with all products currently in the simulation being available in unlimited quantities.

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

The product availability vector $Av_{pos}(t)$ of pos ($Av_{pos}(t) = (av_p^{pos,t}) \in \mathbb{B}^{\mathcal{N}_P}$, indicating whether the respective products p are available at pos at the time t),

The POS price vector $PPr_{pos}(t)$ of pos $PPr_{pos}(t) = (PPr_p^{pos,t}) \in \mathbb{R}_{\geq 0}^{\mathcal{N}_P}$ at time t, relative to a reference price),

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

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

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

1.3.1 Product availability

The availability of a product is modeled through a binary variable for each product and POS $Av(t)$, indicating whether the product is available at the respective pos. For products of a nature where immediate availability / stock management is relevant (which fall out of the scope of this framework), these need to be incorporated through an additional mechanism not considered here.

[2] That is the authority for decision making / the implementation of a decision making procedure for the behaviour of a POS.

1.3.2 Purchase process scheme

The purchase process $PuPS_{pos}$ allows to model how products are purchased at a POS. These are highly linked to decision procedures, and will be employed by these.

Since the purchase process employed in very heterogeneous product groups (such as energy generation technologies and energy rates) is very different, purchase process schemes allow for employing different purchase processes for different product groups, realizing processes / modeling of product purchase.

1.4 Company Agents

IRPact requires a company agent to take management decisions and carry out product management (as manipulation of supply), as well as advertisement, implemented through communication from company to consumer agents. Furthermore they should exhibit the ability for marketing, i.e. to manipulate product attributes.

Company agents represent decision makers or customer interaction entities within companies/firms or nonprofit entities (such as local energy service providers). Based on the requirement analysis, they should allow for behavior and dynamics such as strategic alignment, suppliers influence and competitor influences.

Company agents thus manipulate the supply side of products and control the market status of products (e.g. market introduction of products or discontinuation of products), operationalizing management decisions and product management. In order to facilitate customer interaction, company agents may also send messages to consumer agents (advertisement). Since POS agents represent the link between consumer and company agents, company agents often interact with POS agents, provided both are part of the model. Since many models in innovation diffusion are much more concerned with the demand side of products (i.e. consumer behaviour), in many models the company agents will be non-existing or trivial.

Through modeling company agents, entities representing companies can influence the simulation as well instead of being passive through scripted events, messages or prizing schemes.

Company agents differ from consumer agents in several ways. They do not assume a physical position and are not part of the social network. Messages from company agents are unidirectional and are directed to all agents of a certain agent group (or several of these).

A company agents $coa \in COA$ (with COA the set of company agents) state at time t is described as:

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

$PP_{coa}(t)$ being the product portfolio of agent coa at time t, describing what products the company agent manages (which should partition the set of products in disjoint sets, i.e. $\bigcup\limits_{coa \in COA} \bigcup\limits_{t \in T} \bigcup\limits_{p \in PP_{coa}(t)} p = P = \bigcup\limits_{t \in T} P_t$ and $PP_{coa}(t) \cap PP_{\hat{coa}}(t) = \emptyset \; \forall \text{coa} \neq \hat{coa}, \forall t$),

$PQMS_{coa}$ is the product quality manipulation scheme, representing how the product attributes of the products in the product portfolio of the *coa* can be manipulated (e.g. the pricing of a product),

MDS_{coa} standing for the management decision scheme *coa* uses. This scheme describes how the products in the product portfolio are managed (e.g. introduction of new products and discontinuation of old ones), and how these decisions are taken by the agent,

their advertisement scheme AS_{coa} formalizing how they advertise the products in their portfolio $PP_{coa}(t)$, i.e. specifying how many messages to send to which consumer agent (groups) at what time with what content,

and ia_{coa} as their information authority.

1.4.1 Product quality manipulation scheme

The product quality manipulation scheme describes how a company agent changes the product attributes of the products in their portfolio $PP_{coa}(t)$, be it to change the price of the product, their ecological characteristic, their compatibility with the infrastructure or whatever characteristic they represent. A company agent can use a range of factors for market evaluation and the subsequent measure derived from it, and a PQMS often contains these, but doesn't necessarily have to do so; PQMS can come in any form or shape the modeler wants them to.

1.4.2 Management decision scheme

The management decision scheme, although similar in notion to the product quality manipulation scheme (PQMS), takes a broader perspective than the PQMS and describes how products in the portfolio are managed. Primarily this concerns the market availability of products, i.e. the market introduction of new products and the discontinuation of existing products. For this, a market research / evaluation stage and a decision stage can be distinguished for practical purposes; However this doesn't necessarily need to be the case and can be done in any desired way. The reader is referred to the user documentation or external documentation for the design of individual schemes.

1.4.3 Advertisement scheme

An advertisement scheme specifies when and how advertisement messages are sent to which consumer agents in the course of the simulation. It formalizes the marketing process directed to customers in the form of company agent consumer-directed messages (as described by the consumer agent directed messages and information and advertisement component). These can be customer agent group specific and dynamic, depending on the temporal scheme. For a more detailed description of the advertisement scheme, see the advertisement component.

1.5 Policy Agent

Despite factually having a passive role or being represented as exogeneous variable or through scenarios, a public actor or policy maker in some form appears in a number of models. The role of the policy agent is to manage existing policies as well as introducing new policies. Policies fall in three kinds of categories, namely product-directed policies, prohibitive policies and consumer-directed policies, which can be formalized through schemes. These policies management schemes can theoretically be chosen independently of one another, but the modeler should take care that synergetic effects show as intended. In order to instill the policies, in addition to exercising the schemes, a policy agent also must observe the market. The PolicyAgent aggregates all regulatory aspects falling in the public sphere, i.e. manipulating values of specific product attributes as policies directed at manufactures and consumers, ruling out products with certain characteristics (through product discontinuation), as well as manipulating agent attributes and preferences. As mentioned before, in order to make decision it should also evaluate the market.

For simplicity, the separation of the actor instilling the policies and the company or consumer agent implementing compliant processes or behavior is removed, and the policy agent includes both of them[3].

The policy agent *poa* can thus be formalized as follows:
$poa = (PPS_{poa}, CPS_{poa}, RPS_{poa}, MES_{poa}, ia_{poa})$, with

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

the CPS_{poa}, as the consumer-directed policy scheme,

RPS_{poa} as the regulatory policy scheme,

MES_{poa}, the market evaluation scheme,

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

1.5.1 Product-directed policy scheme

The product-directed policy scheme represents the policies that lawmakers and corporate agents employ for controlling the quality of products. Often these represent ecology-directed measures to control for certain product attributes, such as wastefulness, toxicity, sustainability etc. Through these policy schemes, product attributes values are manipulated, often in a coupled manner. This could occur for example when products violate ecological thresholds and have (for example) to reduce the product attribute representing pollution, on the cost of another product attribute (e.g. price).

[3]A more realistic, albeit over-complicated model would have the policy agent issue the policies and the respective actor implement them, with the policy agent evaluating the implementation. For model dynamics however this is irrelevant, which is why all these mechanics are situated within the policy schemes, with the policy agent.

1.5.2 Consumer-directed policy scheme

Similar to the product-directed policy schemes, the consumer-directed ones describe what policies are used by the policy agent for influencing consumer agents. These represent campaigns, laws and incentives to influence consumer attitudes and behaviors. Formally they influence perceptions and preferences, but are not necessarily limited to this. In theory they could also be used to influence the decision processes used by the agents, manipulate the way consumers get informed (i.e. interact with the information scheme) and could also be used for other purposes if desired by the modeler.

1.5.3 Regulatory policy scheme

The regulatory policy scheme describes the way products are evaluated towards health, safety and environmental standards. It is used to evaluate which products need to be taken of the market, formalizing restrictions on products. The regulatory policy scheme integrates legal acts and the corresponding response of company agents in discontinuing products that don't conform to said standards. In terms of model mechanics, this translates to the scheduling of product discontinuation events of products of certain characteristics (i.e. characteristics that violate certain standards).

1.5.4 Market evaluation scheme

The market evaluation scheme describes the perceptive end of the policy agent. It describes how the policy agent derives information about the market and its actors for other activities, and can be used as auxiliary scheme for other schemes. This allows for imperfect knowledge and internal political mechanisms, as well as for models with perfect transparency.

2 Product Modeling

Products need to be able to be grouped into classes of similar aspects, most of all those that describe certain properties. This is done by introducing the concept of product groups, and by assigning product attributes (resp. product group attributes) to them.

Another requirement is the dynamics of the products, an aspect that will not be discussed in this section however, but in (9), since, as mentioned above, model dynamics are realized through dynamic parts of the model and products are a (static) model entity.

Further it is required that infrastructure constraints on products (prerequisite products and exclusive ones) should be exhibited, standard products should be able to be defined and that products should be able to overwrite the decision process actors employ. These aspects are further discussed in the following.

2.1 Products

Products are modeled as the entities of adoption. They comprise a set of product attributes, describing various qualities of products, are organized in Product-Groups and are further described by their status and longevity within the simulation. Just as AgentGroups, product groups serve to bundle together products with similar properties. Products can be parameterized either through stochastic product initialization (meaning that product attributes are 'randomly' assigned (using a probability distribution)) or can be configured as fixed products with set values. A product $p \in P$ in the set of products P is defined as a 4-tupel $p(t) = (PA_p(t), pga_p, pas_{p,t}, \mathscr{PL}_p)$ with

its set of product attributes $PA_p(t)$ at time t with entries $pa(t) = (pav_{pa,t}, pam_{pa}, pao_{pa}) \in PA_p(t)$ (see (2.2) for more details),

the product group $pg = pga_p = \mathfrak{pgam}(p)$ p is a member of (with $\mathfrak{pgam}$ being the Product Group Association Map: $\mathfrak{pgam} : P \to PG$, associating the products with their respective product group),

its product activation status $pas_{p,t} = \mathfrak{apm}(p, t)$, indicating whether the product is 'active' in the market at a given time, where active means they are introduced to the market already and are not (yet) discontinued ($\mathfrak{apm} : P \times T \to \mathbb{B}$),

$\mathscr{PL}_p$ being the product lifetime distribution of the product. This distribution specifies how long a product can be used upon adoption before it has to be readopted. Upon adoption, the lifetime of an individual adopted product will be drawn from this distribution, and upon expiration of the lifetime of the individual adopted product, it will 'no longer be adopted anymore'. How this is dealt with is also managed by the product readoption scheme.

2.2 Product Attributes

Product attributes $pa(t) \in PA$[4] 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. The values product attributes take on in these quality dimensions are thought to be the true/objective values of the product attribute, and are described through the product attribute value map $\mathfrak{pavm}(pa, t)$ at simulation time t.

In addition to the objective value of the product attribute pa ($pav_{pa,t}$), a product attribute is described by the mutability (pam_{pa}) and observability (pao_{pa}) of the respective attribute. These are determined through the product attribute maps $\mathfrak{pamm}(pa)$ (for the mutability of the product attribute) and $\mathfrak{paom}(pa)$ (for its observability), which are described in detail below.

A product attribute $pa(t) \in PA$ over time can thus be described as a 3-tupel

$pa(t) = (pav_{pa,t}, pam_{pa}, pao_{pa})$ with

$pav_{pa,t} = \mathfrak{pavm}(pa, t)$, being the product attribute value of pa at time t,
$pam_{pa} = \mathfrak{pamm}(pa)$, being the product attribute mutability of pa,
$pao_{pa} = \mathfrak{paom}(pa)$, being the product attribute observability of pa.

[4]The set of product attributes is denoted by PA.

For models where the product attributes have an inherent importance associated with them, a product importance map $\mathfrak{pim} : PA \rightarrow \mathbb{R}_{\geq 0}$ can be used. This however is situated with the respective decision processes, and not part of the product modeling.

2.2.1 ProductAttribute mutability

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 (such as actions by company agents) or not. This is introduced since for some products it might be of interest to have control over the change of product attributes. It is to be expected that the modeler wants some 'soft' product attributes such as its 'sexyness' to be mutable, whereas 'hard' attributes such as 'peak output' should not be mutable. Two clear examples that differ in their mutability are product attributes based on marketing activity and technical parameters. Another example of a mutable product attribute is prize, enabling the modeler to investigate different prizing schemes.

This mechanism is realized through the product attribute mutability map $\mathfrak{pamm}(pa) : PA \rightarrow \mathbb{B}$. It is thus modeled as a boolean value, indicating whether a product attribute can be changed or not. If the product attribute mutability is set to true, product attribute manipulation mechanisms can manipulate it; if set to false, it can't be changed in the course of the simulation.

Thus the product attribute mutability has to be set at the instantiation of the simulation.

2.2.2 ProductAttribute observability

The observability of product attributes describes how observable the quality represented by the product attribute pa is through a numerical value $pao_{pa} \in [0, 1]$, describing in how far its true quality can be assessed by actors, where a value of 0 means that the attribute is not observable at all and a value of 1 means that the value is perfectly observable. It is modeled through the product attribute observability map $\mathfrak{paom}(pa) : PA \rightarrow [0, 1]$. In products where (objective) information is difficult to acquire for the agent, the observability will be low. Product attributes with high observability are product attributes that can be easily (objectively) observed by a consumer.

Observability plays a role in the evaluation of products, when consumer agents can interact and assess a product, such as in the post-purchase evaluation.

Obviously observability is dependent on the modeling of product attribute perception, since without subjective product attribute value perceptions, attribute values can't be modified anyways, and product observations become pointless.

2.3 Product Groups

As mentioned above, product groups serve to bundle together products with similar properties, sharing ProductGroup relations, needs covered by products and (potential) decision overwrite mechanisms.

Analogously to Consumer Agent Groups, product groups serve to enable the modeler to balance homogeneity and heterogeneity through the use of probability distributions for describing their attributes, and to specify relations between these.

A product group $pg \in PG$ is formalized as a 8-tupel

$pg = (PGA_{pg}, PPG_{pg}, EPG_{pg}, FP_{pg}, pgn_{pg}, SP_{pg}, odp, \mathscr{PL}_{pg})$, with

The set of product group attributes associated with pg: $PGA_{pg} \subset PGA$ (see (2.4) for more information),

The set of prerequisite product groups for the product group pg: $PPG_{pg} \subset PG$) (see (2.6.1) for more details),

The excluding product groups $EPG_{pg} \subset PG$ describing which product groups are exclusive for pg (see (2.6.2) for more info),

FP_{pg} as the set of fixed products of product group pg (see (2.5) for more on fixed products),

The product group need $pgn_{pg} = \mathfrak{pgnm}(pg)$ that indicates which needs a product group fulfills (with the product group need map $\mathfrak{pgnm} : PG \to Pot(N)$),

The standard product SP_{pg} for the product group pg (a product p of the product group $(\mathfrak{pgam}(p) = pg)$),

The overwrite decision process odp (with $odp = \mathfrak{odpm}(pg)$, where $\mathfrak{odpm} : PG \to P$ is the overwrite decision process mapping; See (2.7) for more about decision overwriting),

and $\mathscr{PL}_{pg}$ being the (default) product lifetime distribution used for products derived from this product group (see product adoption for more details). For fixed products, this can be overwritten, when products from a product group should have different product lifetime distributions.

2.4 ProductGroupAttributes

Since product group attributes formalize the qualities of product groups, they are the most fundamental aspect of product groups. As mentioned in (2.2), product attributes are understood as qualities of products, and 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. Due to the similar nature of ProductAttributes derived from the same ProductGroupAttribute however, their other aspects (i.e. mutability and observability) are equal over all ProductAttributes derived from this ProductGroupAttribute.

The mapping between the product group attributes and the attributes belonging to them is done through the (bijective) product group attribute product attribute mapping $\mathfrak{pgapam}_{pg} : PGA \to PG_p$, with $p = \mathfrak{pgam}^{-1}(pg)$.

With the exception of fixed products, when a product is instantiated, a value is drawn from the value probability distribution for the respective quality of the product.

Analogous to product attributes, a product group attribute can be formalized as a 3-tupel

$pga = (\mathscr{PGAV}_{pga}, pgam_{pga}, pgao_{pga})$ with

$\mathscr{PGAV}_{pga}$ being the distribution the product attribute values are drawn from via the random variable $X_{PGAV_{pga}}$,

$pgam_{pga} \in \mathbb{B}$ being the product attribute mutability (i.e. whether the value of the corresponding ProductAttribute may change),

$pgao_{pga} \in [0, 1]$ being the product attribute observability (i.e. how easy the true value of the ProductAttribute can be derived) by a ConsumerAgent.

Since the mutability and observability were already described earlier, only the product group attribute value will be discussed here in a little more detail.

2.4.1 Product group attribute value

Describing the essence of product attributes, the product group attribute value distribution $\mathscr{PGAV}_{pga}$ is of paramount important for the modeler, since with this she can directly influence the distribution of the qualities of the products associated with the respective product group. The values of the pa are realizations of the random variable associated with $\mathscr{PGAV}_{pga}$, $X_{PGAV_{pga}}$.

2.5 Fixed Products

As mentioned above, products can be parameterized in two different ways. In the 'default' case, products are generated (through a stochastic process, as realizations of random variables) and values are assigned based on an associated probability distribution. For investigating the adoption of products with known/fixed product attribute values however, these can be instantiated as fixed products.

For these products, in contrast to stochastically generated products, exact numerical values are assigned to the values of the product attributes. For other aspects of product attributes (observability and mutability), these fixed product attributes are associated with product attributes.

For consumer agents, existing fixed products are parameterized through the fixed product awareness distribution $\mathscr{FPAD}^{fp}_{cag}$ (describing how awareness about fp is initially distributed in the consumer agents of group cag) and the initial fixed product configuration distribution $\mathscr{IFPC}^{fp}_{cag}$ (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:

$$fp(t) = (PA_{fp}(t), pga_{fp}, pas_{fp,t}, \mathscr{PL}_{fp}) \in FP = \bigcup_{t \in T} FP_t,$$

with FP_t being the set of fixed products at time $t \in T$ and FP the fixed products over the course of the simulation.

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

Fixed products also serve another purpose. They are used for parameterizing scripted events. These are used to model the market introduction or discontinuation of a product. Since it is assumed, in these cases, that the effect of concrete, fully determined products on the model dynamics is of interest, the products these events refer to are parameterized as fixed products.

2.6 ProductGroup Relations

Another relevant aspect of product groups is the relation between different product groups. This concept enables the modeler to specify restrictions on the possibility of product adoption with regards to the adoption of products of other product groups. The modeling framework implements two restrictions: prerequisite product groups and excluding product groups.

In some constellations of products / product groups, the possibility for adoption of a product might depend on the adoption status of other products. This might be since the adoption of a product is unfavorable if another product has already been adopted or if the adoption of a product is the precondition for the adoption of another product (e.g. gas boilers requiring installed gas pipes). In order to model product dependencies, preconditions of products are considered. These are considered / situated in the decision procedures, and are modeled as a product dependency graph. Products whose preconditions are not fulfilled or that are excluded by other (already adopted products) should not be considered in the decision procedures[5].

2.6.1 Prerequisite product group relations

Prerequisite product group relations (modeled as PPG_{pg} for product group pg) represent the concept that the adoption of a technology is necessary before another technology can be adopted (the adoption of the other technology is a prerequisite for the adoption of the technology of interest). Having a non-empty set of prerequisite product groups means that a product from the (requiring) product group can not be adopted by consumer agents if the agent has not adopted at least one product of each product group listed in the set of prerequisite product groups. These product groups thus represent the set of (all) necessary technologies for the adoption of the product group of interest.

2.6.2 Excluding product group relations

Excluding product group relations (modeled as EPG_{pg} for product group pg) represent the concept that the adoption of a technology is made impossible / forbidden when another technology is adopted. Excluding product groups are thus product groups that are excluded from adoption through the adoption of products of this product group, and thus prevents a consumer agent from adopting any product belonging to a product group in this set. Examples for these are

[5] But again, this depends on the decision procedure taken; If a modeler wants to consider these products for adoption, the framework will not stop them, since it tries to be agnostic about product adoption

products of different product groups that however occupy the same resources. Though similar to same need satisfaction (two product groups satisfying the same needs indirectly exclude one another), this mechanism explicitly models exclusive relations between product groups that cannot (semantically sensibly) modeled by exclusive needs.

2.7 Decision Overwrite

Although in the usual case, the decision process agents use is situated with the corresponding agent group (and only there), some models might have a need for 'overwriting' the decision process. This could occur for example when different agent groups employ decision processes of different cognitive involvement, but some products require a stronger involvement of cognitive factors in the decision process.

This model mechanic can be modeled by the decision overwrite property of product groups (as $odp_{pg} = \mathfrak{odpm}(pg)$, if defined). If a decision overwrite property is set (to a valid decision process), agents will use the decision process specified instead of the decision process associated with their agent group. However, if different product groups with differing decision overwrite settings enter a decision process (and are thus competing/ambiguous), the 'default' decision process of the agent will be used. As such, the modeler should take care to limit decision process overwrite (e.g. through need modeling) to product groups disjoint in decision processes.

3 Perception Modeling

Since IRPact was written specifically with models involving 'non-rational' actors in mind and flexibility is the major focus of this modeling environment, the possibility to allow for cognitive modeling is of major importance. Important aspects of cognitive modeling are (among others) incomplete information, cognitive distortions and (subjective) perception. In order to allow the modeler to model this, the information about product attribute values of actors is based on perception instead of the true value of the product attribute (i.e. $pav_{pa,t}$ for product attribute pa). Perception in this sense means that the actor is not aware of the true value of a product attribute, but has a mechanism to determine the perceived value of a product attribute associated with, coined a perception scheme.

Perception in IRPact is modeled in a two-tiered fashion; While its sufficient to describe the perceptions of the values of the product attribute pa for consumer agent c at time t are described through the perceived product attribute value map $\mathfrak{ppavm} : C \times PA \times T \rightarrow \mathbb{R}_{\geq 0}$[6], it is also advantageous to have a tool to describe how these come about and to understand their temporal dynamics. The modeler might want to abstract some of the technicalities of the $\mathfrak{ppavm}$ away, and describe the dynamics of similar attributes in a common way. For this IRPact offers perception schemes $ps \in PS$ describing the initial status of the $\mathfrak{ppavm}$, i.e. $\mathfrak{ppavm}(c, pa, 0)$, and the temporal dynamics (i.e. $(\mathfrak{ppavm}(c, pa, t) | \mathfrak{ppavm}(c, pa, \hat{t}), \hat{t} < t)$).

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

The initial status of the $\mathfrak{ppavm}$ can be described through the perceived product group attribute value distributions $\mathscr{PPAV}_{cag}$ associated with consumer agent group cag ($\mathscr{PPAV}_{cag} = \bigcup_{pga \in PGA} \mathscr{PPAV}_{cag}^{pga}$). However, perception schemes can use different ways to set up their initial status.

The association of the product group attributes and the respective perception schemes is formalizes through the product perception scheme mapping $\mathfrak{ppsm}_{cag} : PGA \to PS$, assigning every product group attribute $pga \in PGA$ the perception scheme $ps \in PS$ governing how the perceptions $\mathfrak{ppavm}(c, \mathfrak{pgapam}_{pg}(pga), \cdot)$ for derived consumers $c \in C$ are governed.

3.1 Product Awareness

Another way to model incomplete product information is through product awareness. Where 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 at all (if a product is on the market). Product awareness is the basis of all product interaction of consumer agents. If a consumer is not aware of a product, she will neither consider the product in/for decision processes, nor will she communicate about it, or employ other model mechanics related to a product. Thus the consumer 'will act as if the product didn't exist'. This however has to be specified by the process model, the decision process and the need development scheme, and is thus not intrinsically coded into the perception module.

Product awareness is modeled through a (binary) function, the product awareness map $\mathfrak{pawm} : C \times P \times T \to \mathbb{B}$, describing whether the consumer actor c is aware of product p at time t ($\mathfrak{pawm}(c, p, t) = \text{true}$) or not (otherwise). It is parameterized at the initialization of the simulation through the product awareness distributions $\mathscr{PAW}_{cag}^{pg}$ (i.e. $\mathfrak{pawm}(c, p, t) = \text{true} \Leftrightarrow$ "$X_{PAW_{cag}^{pg}}(\omega) = \text{true}$") and will also be achieved when a product is encountered (meaning c encountered p at time $\hat{t}$ $\Rightarrow \mathfrak{pawm}(c, p, t) = \text{true}\ \forall t \geq \hat{t}$). This can be achieved through communication about the product by another consumer agent, marketing campaigns (e.g. modeled as company originated consumer agent messages) or by encountering a product in a POS. Again, how exactly that will be realized depends on other components and the overall configuration of the simulation.

4 Preference Modeling

Many agent-based ID models introduce heterogeneity by ascribing different importance to values, on the basis of individual agents or whole groups. Additionally, a product attribute preference mapping is needed in order to translate these into operational evaluations of products in order to drive the product adoption process.

In IRPact, preferences represent the predilections and aversions of actors in a quantitative manner. These are modeled through *Values* and the importance (strength) actors assign to them, as a numerical value. *Values* are the objects representing moral / ethical dimensions of consumer actors. A high numerical value (strength) assigned to the *Value* of the actor is thought to represent a strong preference, whereas a lower value implies that the actor does not assign importance to the Value. The semantics of these however depend on the objects and processes chosen for the model at hand.

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.

They thus allow for heterogeneity within and between Consumer Agent Groups and are thought to play an integral part in decision making. This however depends on the schemes used in the models, and again the semantics depend on the choices of the modeler.

4.1 PAP Mapping

As mentioned above, perception modeling in this context needs to relate consumer preferences to qualities of products in order to allow for product evaluation within product adoption decisions. For many models, the product attributes and the preferences of consumer agents will not coincide directly. Consumer preferences often stand for (moral) values or goals, whereas product attributes are quantifiable properties of the product. Since a product attribute can correspond to several preferences, and several attributes can be directed to a preference, a weighted map to relate product attributes to preferences is needed. In IRPact, this is operationalized through the *product attribute value preference mapping* $\mathfrak{pavpm}$, which weigh the importance of product attributes to the decision process, if the decision process utilizes them.

Through this mapping, 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. Product attributes are thus indirectly linked to decision processes via preferences.
Formally the mapping is expressed as $\mathfrak{pavpm} : PA \times V \to \mathbb{R}_{\geq 0}$. A high value in this map means that the product attribute and the value are strongly coupled; a low one means they are rather independent. How this is interpreted semantically however depends on the decision processes chosen in the model.

This mapping is selected at the initialization of the simulation and remains fixed throughout the simulation, with the potential exception of the market introduction of new products, where a new mapping is added. This mapping however needs to be specified before the initialization of the simulation.

If this aspect should not be included in the model, the product attributes need to coincide with the consumer agents preferences (it is assumed that the preferences relate to the product attributes directly), and the mapping would be

$$\mathfrak{pavpm}(pa, v) = \begin{cases} 1 & pa \sim v \\ 0 & \text{else} \end{cases},$$

where ˜ means that v is the corresponding value for pa.

Any preferences / product attributes that are not matched will have no effect in the decision making process.

5 Needs Modeling

An agent-based ID framework needs to incorporate a motivation for an adoption decision process, the satisfaction of this impulse through products, and a mechanism to weigh the importance or intensity of these. IRPact does this through the concept of *needs*, which represent the motivation of actors to engage in product adoption. Since products satisfy needs, consumer agents strive to adopt them when needs arise. How needs $n \in N$ arise is described through a need development scheme associated with the ConsumerAgentGroup, allowing for heterogeneous mechanics based on the characteristics of the respective agent group. When products 'expire' (i.e. exceed their live time), the need development scheme is triggered, which will eventually decide how this is tackled. As noted in (2.3), every ProductGroup is associated with a set of needs they satisfy, and only products whose product group satisfies the current need enter the actor's decision process.

5.1 Need Development Scheme

As noted above, a need development scheme describes how needs arise, and give meaning to the needIndicator. They are associated with ConsumerAgentGroups, and are (generally) invoked by the process model to create a list of needs (ordered by the order in which they are to be processed). Their internal mechanics can vary greatly, and more thorough descriptions of need development schemes can be found in the user documentation. Need development schemes need to address what happens with products if their lifetime is exceeded (i.e. it becomes unusable and the associated need is not satisfied anymore), when previously satisfied needs are not fulfilled and (in conjunction with the adoption replacement scheme), how need satisfaction is mediated with the replacement of adopted products when products are discontinued.

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 (ConsumerAgentGroup-specific) NeedDevelopmentScheme. Its interpretation is specified through the *need development scheme* NDS_{cag} (and the reader is referred to the user documentation), but is thought to represent the strength of the need.

6 Decision Modeling

IRPact recognizes the decision process to be a central component of agent-based ID models. As mentioned there, the decision process needs to depend on the consumer, available products, time and potentially more elements, and come in a multitude of ways, as well as being related to product adoption or the adopted products.

Since decisions are rarely modeled for any other type of actors than consumers, this section will only describe the product adoption decisions of ConsumerAgents. Other forms of decision making are found in the schemes of the respective agent types.

Consumer agent make decisions about product adoption based on distinct decision processes. Each actor has a decision process associated with it (through the agent group), representing the cognitive processes the actor employs for deciding for a product. The decision depends on the suitable products, the ConsumerAgent (and thus indirectly the ConsumerAgentGroup the actor is part of), and the time advanced since the beginning of the simulation, formally through the product adoption decision map $\mathfrak{padm} : C \times Pot(P) \times T \times D \rightarrow P$, $\mathfrak{padm}(c, P_{pot}, t, d) = p$, describing the taken product adoption for product p of actor c with the potential (i.e. eligible) products P_{pot} at time t for decision process d. Many decision processes use the state of actors in the social network of an agent or other system aspects as well (and could be formally described as the extended product adoption decision map: $\mathfrak{epadm} : \mathfrak{M} \times T \rightarrow P$, with $\mathfrak{M}$ representing the model as a tupel of all its components); however this depends on the implementation of the decision process and is not inherent in the system design. For individual implementations, the reader is again referred to the user documentation.

Usually the decision process is triggered when a need event is processed and is thus governed by the process model. However, theoretically decision processes can also be executed at another time depending on the modeled system dynamics.

6.1 Adoption

Adoption describes the purchase/installation/usage of a product by a *ConsumerAgent* adopting this product. Each consumer agent has a set of adopted products $ap_{c,t} = \mathfrak{pam}(c, t)$ associated with it. These represent that the actor possesses (or has access to) the corresponding products. From the perspective of the model, this means that needs corresponding to these products (more exactly the corresponding ProductGroup) are satisfied.

The adoption of products is governed by the process model, and thus specified in the respective section. However, it is generally the case that product adoption is the result of decision processes, which decide for the best product (for the actor at that time), which in turn adopts this product in order to fulfill a need.

Products only count as being adopted for the lifetime of the product adoption. The lifetime of a product is drawn from the respective distribution $\mathscr{PL}_{pga}$ at adoption time, and formalized by the partial product attribute lifetime map $\mathfrak{palm} : C \times T \times P \rightarrow \mathbb{B}$, 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:

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

, 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 $x^{pl}_{p,c,t}$ being the realization of the random variable $X^{c,t}_{pl}$ for product p adopted by consumer c at time t, specifying how 'long this product will last'.

[7]To not make it overly complicated, although technically wrong, the author 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.

Adopted products also relate to the products entering the decision process in the case of *product group relations*, in the way that they prevent (exclude) the adoption of certain products or enable the adoption of certain products (prerequisite) (see (2.6)).

6.2 Decision Schemes

As with a range of other aspects in IRPact, the decision process aims to be highly flexible. As such, many mechanisms of the decision processes are abstracted away in their respective schemes. From the perspective of the process model, agents decide based upon their internal state and a limited number of environmental parameters when they are triggered to do so (e.g. by need events). As mentioned above, this can be formalized as the product adoption decision map $\mathfrak{padm} : C \times Pot(P) \times T \times D \to P$, $\mathfrak{padm}(c, P_{pot}, t, d) = p$, with d fixed for the decision scheme used, or more widely through the extended product adoption decision map $\mathfrak{epadm} : \mathfrak{M} \times T \to P$) Concrete decision processes are described in the user documentation and the interested reader is referred to this for more details.

Attitude Importance

In addition to specifying the different consumer agent groups for different product groups, some models are interested in the importance of attitude for an agent group about different product groups. Since this is not integral to consumer agents and links with product groups, this is done separately from them through the attitude importance maps $\mathfrak{aim}_{cag} : PG \to [0, 1]$ on the unit interval. How this attitude is interpreted depends entirely on the decision process, and the map only serves to give a numerical value to the importance of attitude for the respective agent group.

7 Social Network Modeling

A social network component is crucial for IRPact. Although the structure of the connection / communication channels between consumer agents (formalized through the social graph) is the most important aspect of the social network, a requirement for changing the edge weight and the structure of the network was identified as well.

The social network describes the connection between actors. In order to achieves the largest generality, the network is modeled as a dynamic directed weighted (multi[8]-) graph with actors as nodes and connections between actors as edges. For models interested in the dynamics of aspects regarding the social network, IRPact allows for temporal changes in the social network (network dynamics). Dynamic aspects implemented in IRPact are change of edge weights and network topology.

It can be used for multiple purposes; The most obvious being communication flow and adoption patterns of neighbours. The mechanics of how the social network influence the model dynamics are described in their representative sections.

[8]Where edges between two nodes are of a certain type, and only one edge of a given type can be between two actors (in each direction).

However, it is used most extensively in communication and decision processes in most models. The structure of the social network depends strongly on the network chosen and is elaborated on in the user documentation.

The social network is thus modeled as a 3-tupel $SN = (G, \mathfrak{w}, \mathfrak{tms})$ with social graph $G = (AN, E)$, the edge weight manipulation scheme EWS and the topology manipulation scheme $\mathfrak{tms}$, whose components will be discussed in the following.

7.1 Social Graph

At the core of the social network is the social graph $G = (AN, E)$, describing the relationships between actors as nodes $r, o \in AN$, through connecting them with directed edges $(r, o, m) \in E$, where each edge is associated with a medium $m \in M$.

Every node corresponds to a consumer agent c, which is formalized through the consumer agent social graph association function $r = \mathfrak{csgaf}(c)$. The connection between nodes is done through directed edges $e = (r, o, m)$ between nodes (from node r to node o) in the graph. The semantics of the media need to be specified by other model components. Examples for these are communication flow, information flow, friend-of-relations etc. The edges are further associated with different interactions, such as information exchange, value exchange, trust, marketing information etc., which (again) need to be specified in other model components. Thus the edges can also represent different qualities of information flow. The modeler needs to ensure that the semantics and formalization of these is consistent with the way communication in their model is incorporated in IRPact. Furthermore are edges in the social graph associated with a weight through the edge weight function $\mathfrak{w}$, which will be discussed more in-depth in (7.2).

7.2 Edge Weight Schemes

In order to allow for dynamic behaviour and heterogeneous edge weights, IRPact permits the weight of edges to vary over time and between one another, accounting for heterogeneous edge weights at the initialization of the simulation as well as changing the edge weights over time in the course of the simulation. Formally this is described by the edge weight function $\mathfrak{w} : E \times T \rightarrow \mathbb{R}_{\geq 0}$.

In order to balance flexibility and manageability of the function, edge weight schemes $ews \in EWS$ are used. As mentioned above, this forms part of the social network, and governs the edge weights of the social graph over time.

The edge weight scheme $ews = (ewis, ewds)$ includes the edge weight initialization scheme $ewis \in EWIS$ and edge weight dynamics scheme $ewds \in EWDS$, and can be thought to be parametrized through these (where the edge weight initialization scheme is invoked during initialization of the social graph (i.e. specifies $\mathfrak{w}(e, 0)$) and the edge weight dynamics scheme (for $\mathfrak{w}(e, \hat{t}), \hat{t} > 0$) steered by the process model / temporal model).

Edge weight schemes describe under which dynamics the edge weights are determined / calculated in the course of the simulation. Possible edge weight schemes are discussed in the user documentation.

7.3 Topology Mutability Schemes

As a third aspect of the social network, the mutable topology 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 $o \in AN$ (i.e. $\mathfrak{tmf}(r, o, t) = \text{true} \Leftrightarrow \exists\, m \in M : (r, o, m) \in E_t$, as the dynamic edge set). This means that new edges can be added to the social network, as well as that edges can be removed. With this the edge set thus depends on the simulation time (unless a static topology manipulation scheme is used), and the social graph is described as $G = (AN, E) = (AN, E_t)$.

Another aspect of the topology manipulation is the possibility of self-reference within the social graph, meaning edges from one nodes 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 can be desired by the modeler or not, and is specified by the topology manipulation scheme.

One example for where self-referencial edges are desired is for self-reinforcements, as noted in [**?**]. A weighting of these edge can then characterize the degree to which an actor relies on self-reference.

8 Communication Modeling

Communication is a very important aspect of ID models and closely related to the social network component. As identified there, the communication component needs to allow for heterogeneous and flexible schemes for communication between consumers and company agents (e.g. in the form of advertisement), such as the number of peers to communicate and when these communicate. Central messages to these communication events are product perception manipulation messages, and advertisement messages.

Communication within IRPact is understood as the generation and interpretation of messages, most prominently through consumer actors. This is generally thought to take place through the social network, but is not necessarily limited to this. The timing of messages (sending and interpretation time) is managed by the process model and the communication scheme CS_{cag} of the respective consumer agent group, as parameterized through the *cag*. The effect of processing a message is described by the message itself and depends on the type of message.

The Communication Scheme CS_{cag} further governs how messages are scheduled in the temporal course of the simulation. Where the message schemes describe what and how many messages are created, the communication scheme determines when these are sent through the social network. This done through modeling them as communication events, by associating the messages with the point in time when they get evaluated. Formally this can be described by the communication event mapping $\mathfrak{cem}_{cag} : COPM_{cag} \times T \to CE_{cag}$[9].

As mentioned before, a message scheme describes how messages of a given type are created, depending on the sender of the message and the state of

[9] While this mapping should be defined over a more generous domain allowing messages other than cop messages, these were the only ones considered for the current state of the framework, and so I believe it would be more confusing than adding clarity to the model to keep this more general

the simulation. It can thus be formalized as a function $\mathfrak{ms}_{cag} : C \times \mathfrak{M} \times T \to Pot(COPM)$[10], describing which messages consumer agents $c \in C$ with $\mathfrak{cagm}(c) = cag$ send at time $t \in T$ depending on the model state $\mathfrak{M}$.

The communication scheme can thus be described as a pair $CS_{cag} = (\mathfrak{ms}_{cag}, \mathfrak{cem}_{cag})$ of the respective message scheme and the respective communication events as the communication event mapping.

Messages originating from consumer agents $copm \in COPM$ are formalized by the *ConsumerAgentMessageScheme*, of which a crucial component is the message activity distribution. The message activity distribution $\mathscr{MA}_{cag}$ of consumer group cag, represents how many messages a consumer agent c of group cag sends to other consumer agents $\hat{c}$ they are connected to in the social graph G per time unit.

Since messages are sent through communication channels (edges in the social network), messages are dependent on the *types of links* m in the social graph. The links do not solely represent communication channels, but can stand for aspects of relationships between agents (such as trust), so communication mechanisms need to take into account the nature/semantics of these links.

8.1 Consumer-originated Messages

At this point, the only relevant type of consumer-originated messages, i.e. messages from a consumer agent to another (neighboring, in the sense of the social graph G) consumer agent are the product perception manipulation messages $copm \in COPM$. This message, representing word of mouth (about a certain aspect (i.e. quality) of a product), changes the (perceived) product attribute values of the receiving agent, depending on the perceived product values of the sender of the message, and can be formalized as a 4-tupel $copm = (c, \hat{c}, pa, p)$, as a message stemming from sender c to receiver $\hat{c}$ about product attribute pa of product p.

If the product p is active within the simulation, the receiver becomes aware of it (i.e. $\mathfrak{pawm}(\hat{c}, p, \hat{t}) = \text{true}\, \forall \hat{t} > t$, with t the time of the message evaluation) if they aren't already, and a new perception of this product attribute is added.

The perception of the product attribute pa the message is based on the perception of the sender c of the message (i.e. $\mathfrak{ppavm}(c, pa, t)$) and its strength depends on the edge weight (i.e. $\mathfrak{w}((r, o, m), t)$, with $r = \mathfrak{csgaf}(c), o = \mathfrak{csgaf}(\hat{c})$).

How this perception is incorporated in 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 $\hat{c}$).

8.2 Company-originated Messages

The way messages are sent from company agents is governed by the companyAgentMessageScheme (see user documentation for details). These schemes regulate how the communication of company agents to consumer agents takes place, and serve to describe the targeting scheme / strategy of companies towards the consumer agents. It acts in conjunction with the advertisement scheme, since many messages of company agents are advertisement messages, which are governed by this scheme.

[10] The same as in the footnote above applies here

8.2.1 Product perception manipulation message (company)

A product perception manipulation message (company) message changes the product attribute value perception of a receiving consumer agent $\mathfrak{ppm}$ through a message sent by a company agent, representing marketing messages. They can also represent model processes such as branding, in which a number of recipients (i.e. agent group) will have some product attributes manipulated.

The modeler should pay attention to the product attribute these messages manipulate in order to make sense semantically. The effect of this message is that the perception of a product attribute of the receiving agent is changed by a certain amount. Exactly how this is done depends on the respective scheme however.

These messages might take into account different links as well, e.g. links that stand for trust of a consumer agents to the company represented by the company agent.

8.2.2 Preference manipulation message (company)

As a more indirect form of product marketing, company-originated messages can also influence the preferences $\mathfrak{pm}_c$ of a consumer agent c[11]. This is done to represent advertising campaigns that stress the importance of values in which the products of the company are particularly strong (e.g. a company with very sustainable products might want to strengthen consumer preference for sustainability in order to make their products more attractive for customers).

However, the modeler should use this aspect of the framework cautiously, since preferences are a major defining factor for actor groups and this could water down the boundaries between the behaviour of actor groups. As with the preference manipulation messages, the number of messages of a given type sent in each step (time interval in the continuous model) is a fixed distribution for each agent group, and selected at the beginning of the simulation.

8.3 Message Schemes

Message schemes govern the mechanics of how messages are generated. Since messages are sent between actors representing cognitive entities (agents), the messaging scheme used depends on the modeling of the agents. Because message schemes vary broadly and implement their own mechanics, the reader is once again referred to the user documentation, and further specifications at this point would remain solely abstract.

9 Event Modeling

The temporal order of processes is important, and the temporal dynamics of the model should be situated in a common framework, in this case the event

[11]Although because of this the preference map $\mathfrak{pm}_c$ of each consumer should be described as a family of functions to be technically correct, this would have added significantly more complexity to the model description. Since preference manipulation messages are much rather a theoretical construct (and as such a very 'dangerous' one, since its introduction makes understanding consumer behavior a lot more difficult), this potential was sacrifices for simplicity and clarity of description

modeling. Events in this understanding are processes occurring at a specific time, bringing together functionality and execution time. Through this they are strongly intertwined with the temporal model and the process model, and thus describe a number of model dynamics.

Relevant events for IRPact are the introduction of a new product, the discontinuation of a product, the communication or transmission of information or advertisement, the surging of a need and the evaluation of an adopted item. These are presented in the following.

9.1 Scripted Events

Scripted events are events that are set to a given execution time $\hat{t} \in T$. They are generally configured through the parametrization of the simulation (i.e. depend on the model $\mathfrak{M}$), and can be formalized by the scripted events scheduling function $\mathfrak{scsf} : SE \to T$.

Currently the only scripted events are scripted product events, which serve to add (in the case of market introduction events) or remove products from the market (in the case of product discontinuation events), but this can easily be extended.

Scripted events are supplied exogeneously to the model with a fixed time signature and must conform to the temporal regime of the system (i.e. must lie in a temporal domain of the clock (see (11))). Scripted events are events that occur at a certain time of the simulation.

9.1.1 Market Introduction Events

Market introduction events represent new products becoming available to the simulation. Although these events represent that new products are introduced to the market, technically they *activate* products, meaning that after this products become eligible to adoption (and potentially other mechanics within the simulation). Formally this means that if product p is introduced at time $\hat{t} \in T$ it holds that $\mathfrak{apm}(p, t) = \text{true} \forall t \geq \hat{t}$, as long as no product discontinuation event occurs.

As scripted events they are set to a certain simulation time, representing the time the products are introduced, and are further associated with fixed products fp, describing the product to be introduced. This can be 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). Even though the events are associated with a fixed simulation time coming from a modeling idea about fixed product interactions, these events could also be dynamically created by company agents deciding on a product to be introduced to the market (when incorporated into the model). The market introduction of products enable actors to interact with them.

The introduction of new products is realized as a *scripted event* that gets determined at model selection. Until market introduction, the product is inactive, i.e. its value in $\mathfrak{apm}$ is specified as *false* until the event occurs.

9.1.2 Product Discontinuation Events

Being the antipole of market introduction events, these events represent the process of making a product unavailable to customers. Consumer agents will not be able to adopt this product anymore, and options for other kinds of actors will be restricted as well. This means that when product p is to be discontinued at time $\hat{t}$ it holds that $\mathfrak{apm}(p, t) = \text{false}\, \forall t \geq \hat{t}$. Product discontinuation events are also related to adoption replacement schemes, since the discontinuation of a product (especially if the product represents a service) may involve a re-adoption of the actor.

The discontinuation of products is realized as a *scripted event* that is determined at model selection time, and the effects of the discontinuation of that product have to be resolved when the event occurs. This necessitates that modules using a product are responsible for checking whether a product is still on the market.

9.2 Communication Events

Since communication can occur in a multitude of fashions, communication events are based on *messages*. Processing a communication events means processing the message the communication events is based on, and is elaborated upon in the section about communication.

9.3 Need Events

Need events are strongly related to needs, and represent the process that occurs when an actor intends to satisfy a need. The scheduling of the need event thus represents the rise of the need, where processing the need event initiates the process the actor engages in to satisfy a need, and allows for a distinct timing of these two processes.

Processing a need event compiles a list of products the consumer agent 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.

Products conforming to these criteria enter a decision process, which is either a commonly overwritten decision process *odp* of the products or the decision process associated with the actor.

9.4 Post-Purchase Evaluation Events

Based on the assumption that using a product gives a consumer actor access to assess the true product attribute values to some extent, post-purchase evaluation events are events that serve to modify the consumer agent's product attribute perception after the adoption of a product.

Upon evaluation, they add a product perception to the consumers product attribute perception with a weight based on the observability of the product and the true product attribute value
(i.e. $\mathfrak{ppavm}(c, pa, t) | (\mathfrak{ppavm}(c, pa, \hat{t}), poa_{pa}, pav_{pa,\hat{t}})$). As noted in the section about product perception schemes, the semantics of this information is left to the product perception scheme, which are elaborated on in the user documentation.

When and how often these events will be scheduled depends on the process model, and needs to be specified there.

9.5 Information Events

As further described in (10.1), information events describe how information is dynamically situated in the simulation. Currently the only information of interest are information about product attributes. Information events thus link together information $i \in I$, the (at this state of the model) consumer agent $c \in C$[12] interpreting it and the time of information evaluation $t \in T$ as a triple $ie = (i, c, t)$.

The semantics of (product attribute) information events is that their evaluation is 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 $\mathfrak{iam} : IA \to \mathbb{R}_{\geq 0}$ information agents exhibit (see (10.1)). Here, IA is the set of information agents as $IA = C \cup POS \cup CA \cup poa \ni c$.

10 Information and Advertisement Modeling

Information sources and channels as well as advertisement are important aspects an ID framework needs to address, particularly how a range of information and advertisement processes take place, such as changing the perception or preference of an agent, taking account the true nature of a product for information and some desired value for advertisement.

Information and advertisement serve to change the perception or preference of consumer agents. The major difference (in this framework) is that information is actively sought out by consumer agents, whereas advertisement is targeted at them by company agents or scripted events, and the consumer agents are thus more passive. Where these processes are situated within the simulation is governed by the process model.

Information seeking is often strongly tied to decision making. It describes how actors seek out information within decision processes or potentially other processes, and is governed by schemes invoked from the process model or the respective decision processes. As with many aspects of the framework, these are realized as schemes, describing what information actors think they require and what strategies they use to find it, which are elaborated on in the user documentation. Due to their diversity, formalization would only serve symbolic meaning and was thus not done.

10.1 Information Modeling

Information in IRPact is modeled from a number of perspectives. Information itself is understood in a naive form, as a piece of knowledge $i \in I$[13] that originates from an information agent, that is an agent that is capable of providing and potentially processing information (see (1) for more details on this). On the

[12]Since information events are only directed at changing product perceptions and preferences, currently the only agents interpreting them are consumer agents, whereas in the future they are thought to be generalized to other information agents as well

[13]for I the set of all information

basis of this and the respective ConsumerAgent $c \in C$ interpreting this information, information events are defined as a triple $ie = (i, c, t)$ that are processed at simulation time $t \in T$.

Interpreting information is currently only specified for product attribute information, which changes the perception of the processing agent according to the perception scheme by acting as a perception respective to the product attribute (which product attribute the perception affects can be specified by the partial information product attribute mapping function $\mathfrak{ipam} : I \to PA$, defined for all product attribute information $pai \in PAI \subset I$).

The way information enter the simulation and are taken up by the agent is governed by information schemes. Formally an information scheme specifies what information is available for agents to process (via the information availability mapping $\mathfrak{iavm}$) and how product information is sought by consumer agents (through the product information map $\mathfrak{pim}$).

The information availability mapping $\mathfrak{iavm} : I \times T \to \mathbb{B}$ indicates whether a given information is 'known' at a given time $t \in T$. It is thus modeled as a boolean function indicating whether the information $i \in I$ is available ($\mathfrak{iavm}(i, t) = \text{true}$) at time t or not (false).

The product information map $\mathfrak{pim} : C \times PA \times T \to \mathscr{P}(I)$ describes 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.

Information schemes thus specify how these model mechanics work and through this describe the information ecosystem within the simulation as a pair $(\mathfrak{iavm}, \mathfrak{pim})$. Since the choice of the information scheme depends on the mechanisms to model, a number of information schemes are found in the user documentation.

10.2 Advertisement Modeling

Advertisement in IRPact is modeled as company originated consumer agent oriented communication messages. Advertisement stems from company agents, through their *advertisement scheme* AS_{coa}. As noted in (1.4), each company agent has associated with them such an advertisement scheme, describing the process of advertising the product portfolio of company agents. This usually includes generating the respective (company originated consumer agent directed) messages and their transformation into events. This can depend on a number of things, and each implementation should note how the marketing department of the respective company agent acts according to the state of the simulation.

Advertising schemes enable the company agent to send advertising messages to agents in the simulation. These messages can be *product perception manipulation messages* or *preference manipulation messages*. Concrete details depend on the implementation and are described in the user documentation, and since these are rare (if extant at all) are not further specified here.

11 Temporal Modeling

Time needs to be an integral part of ID models, especially to allow for model dynamics. As mentioned there, a more general framework with the potential for different timing schemes is needed.

This is done with the temporal model, describing how time is represented within the model. 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. It is hierarchically situated above the process model, and drives the process model within the temporal frame. A temporal regime further includes the functionality for handling scripted events.

Different time models allow the system to follow different timing regimes. This comprises the discrete and continuous time model, but should be set out to allow for other time models as well. In general, discrete time models operate in steps that proceed 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 usually 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. A more thorough description of these models can be found in the user documentation.

12 Spatial Modeling

Spatiality is an important characteristic in a range of models. It 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, implying the use of 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 (and by this closeness) are understood.

The spatial model describes the spatial aspects of the simulation from a relational perspective. Whereas the positioning of the (spatial) actors is done through their coordinates loc, their positioning towards the geometry of other spatial model aspects such as the model border (by the topographic scheme) and one another (by the metrical scheme) is described by the spatial model $\mathfrak{S} = (TS, MS)$.

Implementations of space schemes serve to situate the (spatially bound) agents more specifically. These can be found in the user documentation.

Entities within the simulation are further associated with a spatial distribution (e.g. $\mathscr{S}\mathscr{D}_{cag}$ for consumer agents) for the geographical initialization of the model. These entities are thus assigned a position in the spatial model according to the distribution.

12.1 Topographic Scheme

The topographic scheme establishes the spatial borders of the simulation and governs how/where agents are situated within the simulation. Since few models represent the topography of the spatial context of their models by more than the borders of simple geometric objects, the core of the topographic scheme deal with determining whether given coordinates lie within the boundaries of the spatial model.

This can be described by the spatial border map $\mathfrak{sbm} : \mathbb{R} \times \mathbb{R} \to \mathbb{B}$, describing whether a coordinate *loc* falls within the borders of the model ($\mathfrak{sbm}(loc) = \text{true}$) or outside ($\mathfrak{sbm}(loc) = \text{false}$).

For all of these simple cases considered here, the topographic scheme can thus be identified with this map, (i.e. $TS = \mathfrak{sbm}$).

12.2 Metric Scheme

For aspects of the model that are based on the distance of actors to other model entities (such as points-of-sale, other actors or geographic features), the metric scheme specifies the metric that is used to determine their distance. Together with the valid location points within the topographic scheme, the spatial model can thus be understood as a metrical space.

Currently only straightforward metrics (1-, p- and ∞- metric) were regarded for model design; with more specific semantics however, other metrics could (and should) be incorporated for the spatial model.

13 Process Modeling

The stages of the innovation decision process (at least with the stages of awareness, trial and adoption) necessitate a process model. This component needs to allow for flexibility and include the execution of dynamic aspects of the model via an event scheduler and contain an adoption replacement scheme, specifying what happens after the discontinuation of a product.

It is supposed to be subordinate to the temporal model and invoked by it, and specify the execution of processes (for discrete temporal models) or steps between events (for continuous temporal models). It thus depends on the implementation of the temporal model. Where 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.

The process model specifies how actors are processed. It is in most cases 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.

With this, the process model could be specified as the pair $PM = (es, ars)$ of the event scheduler $es \in ES$ and the adoption replacement scheme $ars \in ARS$.

13.1 Event Scheduler

As mentioned in (9), every event in IRPact (such as decision making, message passing etc.) is assigned a certain point in time when its evaluated. While their execution is more an implementational detail that will be described in (**??**), this could be formalize by the specification of the event scheduler $es \in ES$, as chosen from the set of event schedulers ES.

Which event scheduler should be chosen for model formulation depends on the temporal model used and the desired behavior. Different event schedulers can easily be implemented (see the technical documentation for more details), and thus the set of event schedulers ES should not be seen as static.

13.2 Adoption Replacement Scheme

The adoption replacement scheme $ars \in ARS$ describes how actors act after a product has been discontinued (taken off the market). In most cases it will create a need event, although this might depend on different circumstances and the nature of the agents adopting the product as well as the corporate agents managing the product, and no restrictions are given to these schemes a-priory. Due to its generality however, this depends strongly on the implementation, and the reader is referred to the user documentation for more details.

As such any further formalization would remain meaningless and abstract.

Model Discussion

As mentioned in the introduction of this chapter, 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 and was very strongly shaped by the components and their respective requirements there.

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

The super agent layer consists of all aspects overarching the agents, 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.

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.

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.

Through this a comprehensive, formally consistent and highly flexible model description was reached, allowing model formulation, development and documentation in a concrete, comprehensive and concise manner.

Nomenclature

$\mathcal{N}_{PA}^{t}$ The being the number of product attributes $pa \in PA$ at time t $\Big(\mathcal{N}_{PA}^{t} = card(PA_t)\Big)$

$\mathcal{N}_P$ The number of products used in the simulation ($\mathcal{N}_P = card(P)$)

$\mathcal{N}_V$ The number of of values used in the simulation ($\mathcal{N}_V = card(V)$)

ia_a The information authority ia_a of an information agent a is a (over the course of the simulation constant) scalar that describes how trustworthy information stemming from a is for other agents. It is of importance in processes of other agents, where information from the agent needs to be evaluated

$\mathscr{CP}_{cag}^{v}$ The consumer agent preference distribution for value v for the consumer group cag: Is used to determine the initial preferences of an agent $c \in \mathfrak{cagm}^{-1}(cag)$

$\mathscr{FPAD}_{cag}^{fp}$ The fixed product awareness distribution of consumer agent group cag for product fp. Describes how the awareness for this product is distributed for consumer agents c of this group ($c : \mathfrak{cagm}(c) = cag$)

$\mathscr{IFPC}_{cag}^{fp}$ The initial fixed product configuration distribution of consumer agent group cag for product fp. Describes how the fixed product is distributed for consumer agents c of this group ($c : \mathfrak{cagm}(c) = cag$)

$\mathscr{MA}_{cag}$ The message activity distribution of consumer group cag, representing how many messages a consumer agent $c \in cag$ sends to other consumer agents $\hat{c}$ they are connected to in the social graph G per time unit.

$\mathscr{PGAV}_{pga}$ The product group attribute value distribution, describing the distribution of the product group attribute value of product group attribute pga

$\mathscr{PL}_p$ The product lifetime distribution of the product, specifying how long a product can be used upon adoption before it has to be readopted. Upon adoption, the lifetime of an individual adopted product will be drawn from this distribution, and upon expiration of the lifetime of the individual adopted product, it will not be adopted anymore.

$\mathscr{PPAV}_{cag}^{pga}$ The perceived product group attribute value distribution for product group attribute pga of the consumer group cag: It is used to determine the initial perception of the value of a product attribute for a number of histogram initialization schemes $pa(0) \in PA_p : PGA_{pg}, pg = \mathfrak{pgam}(p)$

$\mathscr{SD}_{cag_c}$ The spatial distribution associated with a consumer agent group corresponding the agent is part of (i.e. $\mathscr{SD}_{cag_c} = \mathscr{SD}_{cag}$ for $\mathfrak{cagm}(c) = cag$)

$\mathscr{SD}_{cag}$ The spatial distribution associated with a consumer agent group used to initialize an agent within the spatial model $\mathfrak{S}$

$X_{pl}^{c,t}$ The random variable for product p adopted by consumer c at time t, specifying how 'long' this product will last

$X_{CP_{cag}^v}$ The random variable $X_{CP_{cag}^v} : \mathscr{CP}_{cag}^v \to \mathbb{R}_{\geq 0}$, representing which initial preference an agent c has for a value v ($\mathfrak{pm}_c(v, 0) = X_{CP_{cag}^v}(\omega)$)

$X_{FPAD_{cag}^{fp}}$ The random variable associated with the fixed product awareness distribution of consumer agent group cag for product fp. Describes how the awareness for this product is distributed for consumer agents of this group ($X_{FPAD_{cag}^{fp}} : \mathscr{FPAD}_{cag}^{fp} \to \mathbb{B}$)

$X_{IFPC_{cag}^{fp}}$ The random variable associated with the fixed product configuration distribution of consumer agent group cag for product fp. Describes how the initial configuration for this product is distributed for consumer agents of this group ($X_{IFPC_{cag}^{fp}} : \mathscr{IFPC}_{cag}^{fp} \to \mathbb{R}_{\geq 0}$)

$X_{MA_{cag}}$ The random variable $X_{MA_{cag}} : \mathscr{MA}_{cag} \to \mathbb{N}_0$, representing how many messages a consumer agent $c : \mathfrak{cagm}(c) = cag$ sends to other consumer agents $\hat{c}$ they are connected to in the social graph G per time unit, depending on the corresponding message activity disctribution $\mathscr{MA}_{cag}$

$X_{PGAV_{pga}}$ The random variable associated with the product group attribute value distribution, describing the distribution of the product group attribute value of product group attribute pga

$X_{PPAV_{cag}^{pga}}$ The random variable $X_{PPAV_{cag}^{pga}} : \mathscr{PPAV}_{cag}^{pga} \to \mathbb{R}_{\geq 0}$, representing which (initial) perceived value an agent c of associated agent group cag has of a product attribute pa of product attribute group pga ($\mathfrak{pavm}(pa, 0) = X_{PPAV_{cag}^{pga}}(\omega)$)

$X_{SD_{cag}}$ The random variable associated with the spatial distribution $\mathscr{SD}_{cag}$ used to initialize an agent within the spatial model $\mathfrak{S}$ ($X_{SD_{cag}} : \mathscr{SD}_{cag} \to \mathfrak{S}$), where in most cases $\sigma \in \mathfrak{S}$ is a set of spatial coordinates

$\mathfrak{aim}_{cag}$ The attitude importance map is a map describing how important attitude is for the agent group cag in decisions involving different product groups $pg \in PG$ ($\mathfrak{aim}_{cag} : PG \to [0, 1]$)

$\mathfrak{apm}$ Active product map, indicating which products are 'active' in the market at a given time, where active means they are introduced to the market already and are not (yet) discontinued: $\mathfrak{apm} : P \times T \to \mathbb{B}$

$\mathfrak{cagm}$ The Consumer Agent Group Association Map: A map indicating which consumer agent group a given consumer agent is a member of ($\mathfrak{cagm} : C \to CAG$)

$\mathfrak{cem}_{cag}$ The consumer event mapping of consumer agent group cag, $COPM_{cag} \times T \to CE_{cag}$, associating each copm of that group and point in time with the communication event it describes

$\mathfrak{csgaf}$ The (bijective) consumer agent social graph association function determines which node in the graph a consumer corresponds to ($\mathfrak{csgaf} : C \to AN$), with AN being the set of nodes in the social graph

$\mathfrak{epadm}$ The extended product adoption decision map: $\mathfrak{epadm} : \mathfrak{M} \times T \to P$, generalizing the $\mathfrak{padm}$ to depend on any aspect within the model $\mathfrak{M}$

$\mathfrak{iam}$ The information authority mapping $\mathfrak{iam} : IA \to \mathbb{R}_{\geq 0}$ describes the authority information agents exhibit, in order to inform functionalities where the authority of an information is relevant. In this, IA is the set of information agents as $IA = C \cup POS \cup CA \cup poa \ni c$.

$\mathfrak{iavm}$ The information availability mapping $\mathfrak{iavm} : I \times T \to \mathbb{B}$ indicates whether a given information is 'known' at a given time $t \in T$. It is thus modeled as a boolean function indicating whether the information $i \in I$ is available ($\mathfrak{iavm}(i, t) = \text{true}$) at time t or not (false).

$\mathfrak{iavm}$ The information availability mapping $\mathfrak{iavm} : I \times T \to \mathbb{B}$ indicates whether a given information is 'known', is around the simulation for a given time $t \in T$. It is thus modeled as a boolean function indicating whether the information $i \in I$ is available ($\mathfrak{iavm}(i, t) = \text{true}$) at time t or not (false)

$\mathfrak{ipam}$ The information product attribute mapping function $\mathfrak{ipam} : I \to PA$, maps all product attribute information $pai \in PAI \subset I$ to the product attribute they inform about, in order to link information and product attributes as a partial function

$\mathfrak{ms}_{cag}$ The message scheme of consumer agent group $cag \in CAG$, $\mathfrak{ms}_{cag} : C \times \mathfrak{M} \times T \to Pot(COPM)$, describes which messages consumer agents $c \in C$ with $\mathfrak{cagm}(c) = cag$ send at time $t \in T$ depending on the model state $\mathfrak{M}$.

$\mathfrak{ni}$ The need indicator function, specifying how strongly a need arises for consumer agent groups ($\mathfrak{ni} : CAG \times N \to \mathbb{R}_{\geq 0}$)

$\mathfrak{odpm}$ The overwrite decision process map indicating which overwrite decision process is used if any ($\mathfrak{odpm} : PG \to D$, as a partial function since not all product groups have a overwrite decision process)

$\mathfrak{padm}$ The product adoption decision map $\mathfrak{padm} : C \times Pot(P) \times T \times D \to P$, $\mathfrak{padm}(c, P_{pot}, t, d) = p$, describing the taken product adoption for product p of actor c with the potential products P_{pot} at time t for decision d

$\mathfrak{palm}$ The product attribute lifetime map describing whether an adopted product is operational (i.e. in a state to satisfy the associated need) at a given time, formalized as $\mathfrak{palm} : C \times T \times P \to \mathbb{B}$

$\mathfrak{pamm}$ The product attribute mutability map: Associates a product attribute with its mutability: $\mathfrak{pamm} : PA \to \mathbb{B}$

$\mathfrak{pam}$ The product adoption map indicates which products are adopted by a consumer agent c at time t: $\mathfrak{pam} : C \times T \to P_{ad,c,t} \subset P$

$\mathfrak{paom}$ The product attribute observability map: Associates a product attribute with its observability: $\mathfrak{paom} : PA \to [0, 1]$

$\mathfrak{pavm}$ The product attribute value map: Associates a product attribute with its value: $\mathfrak{pavm} : PA \times T \to \mathbb{R}$, which is a function constant in its second argument when the product is not mutable (i.e. $\mathfrak{pamm}(pa) = 0$)

$\mathfrak{pavpm}$ Product attribute value preference mapping: specifies how strong a product attribute corresponds to a value: $\mathfrak{pavpm} : PA \times V \to \mathbb{R}_{\geq 0}$. A high value in this map means that the product attribute and the value are strongly coupled; a low one means they are rather independent

$\mathfrak{pawm}$ Product awareness map $\mathfrak{pawm} : C \times P \times T \to \mathbb{B}$, describing whether the consumer actor c is aware of product p at times t ($\mathfrak{pawm}(c, p, t) = \text{true}$) or not (otherwise).

$\mathfrak{pgam}$ Product group association map: The map indicating which product group a product p is an instance of ($\mathfrak{pgam} : P \to PG$)

$\mathfrak{pgam}$ The Product Group Association Map: A map indicating which product group pg a given product p is a member of ($\mathfrak{pgam} : P \to PG$)

$\mathfrak{pgapam}_{pg}$ The mapping function between the product group attributes pga and the product attributes pa belonging to them as a (bijective) function $\mathfrak{pgapam}_{pg} : PGA \to PG_p$, with $p = \mathfrak{pgam}^{-1}(pg)$, the product group attribute product attribute mapping.

$\mathfrak{pgnm}$ The product group need map: Indicates which needs a product group fulfils ($\mathfrak{pgnm} : PG \to Pot(N)$)

$\mathfrak{pim}$ The product importance map $\mathfrak{pim} : PA \to \mathbb{R}_{\geq 0}$ specifies for each product attribute (or a subset of them if used as a partial function) how important that product attribute is for the product. The semantics of this however are situated within the respective decision process.

$\mathfrak{pim}$ The product information map $\mathfrak{pim} : C \times PA \times T \to \mathscr{P}(I)$ describes what information about a product attribute $pa \in PA$ are relevant to a consumer agent $c \in C$ at time $t \in T$ in order to derive relevant information events

$\mathfrak{pm}_c$ The preference map describes the preferences of agent c: $\mathfrak{pm}_c : V \times T \to \mathbb{R}_{\geq 0}$

$\mathfrak{ppavm}$ The perceived product attribute value map as a function $\mathfrak{ppavm} : C \times PA \times T \to \mathbb{R}_{\geq 0}$, with $\mathfrak{ppavm}(c, pa, t)$ representing the perception of the value of product attribute pa the actor c has at time t.

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

$\mathfrak{sbm}$ The spatial border map $\mathfrak{sbm} : \mathbb{R} \times \mathbb{R} \to \mathbb{B}$, describes whether a coordinate loc falls within the borders of the model ($\mathfrak{sbm}(loc) = \text{true}$) or outside ($\mathfrak{sbm}(loc) = \text{false}$)

$\mathfrak{sesf}$ The scripted events scheduling function $\mathfrak{sesf} : SE \rightarrow T$, describing at what time $t \in T$ a scripted event $se \in SE$ is evaluated

$\mathfrak{speam}$ The scripted product event association mapping $\mathfrak{speam} : SE \supset SPE \rightarrow FP$, associates the scripted product events as subset of the scripted events with the fixed products $fp \in FP$ they are associated with

$\mathfrak{tmf}$ The topology manipulation function $\mathfrak{tmf} : AN \times AN \times T \rightarrow \mathbb{B}$, indicating whether at time $t \in T$ an edge exists between node $r \in AN$ and node $o \in AN$. Is used by a topology manipulation scheme $tms \in TMS$

$\mathfrak{w}$ The social graph edge weight function $w : E \times T \rightarrow \mathbb{R}_{\geq 0}$, assigning the edges edges within the social graph their weight at time $t \in T$

$csgaf_c$ The consumer agent social graph association $csgaf_c = \mathfrak{csgaf}(c)$, describes which node r in the social graph corresponds to c

AS_{coa} The advertisement scheme AS_{coa} formalizes how a company agent coa advertise the products in their portfolio $PP_{coa}(t)$, i.e. specifying how many messages to send to which consumer agent (groups) at what time with what content

CPS_{poa} consumer-directed policy scehemes CPS_{poa} describe what policies are used by the policy agent for influencing consumer agents, e.g. influencing perceptions, preferences, decision processes, information schemes among others.

CS_{cag} A communication scheme governs how messages are scheduled in the temporal course of the simulation, and is defined as $CS_{cag} = (\mathfrak{ms}cag, \mathfrak{cem}_{cag})$, a pair of the respective message scheme and the respective communication events as the communication event mapping.

ews The edge weight scheme $ews = (ewis, ewds)$ includes the edge weight initialization scheme $ewis \in EWIS$ and edge weight dynamics scheme $ewds \in EWDS$, and describes the change of edge weights within the social graph

MDS_{coa} The MDS_{coa} describes the management decision scheme the company agent coa uses. This scheme describes how the products in the product portfolio are managed, and how these decisions are taken by the agent

MES_{poa} the market evaluation scheme MES_{poa} describes the perceptive end of the policy agent, i.e. how the policy agent derives information about the market and its actors for other activities.

MS The metric scheme MS determines the (metric) interpretation of the positioning of spatial agents towards one another (implements a metric, so metrical spaces can be represented)

PPS_{poa} The product-directed policy scheme PPS_{poa} represents the policies that lawmakers and corporate agents employ for controlling the quality of products (when product attributes of which products are manipulated)

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

RPS_{poa} The regulatory policy scheme RPS_{poa} describes the way products are evaluated towards health, safety and environmental standards, scheduling of product discontinuation events of products of certain characteristics.

TS The topographic scheme TS describes the positioning of spatial agents towards the geometry of other spatial model aspects (such as the model border)

a An information agent, $a \in IA = C \cup POS \cup CA \cup poa$

ars An adoption replacement schemes $ars \in ARS$ that can be used in the process model

c A consumer agent as a member of the set of consumer agents ($c \in C$). The consumer state at time t can be described a 9-tupel
$c_t = (cag_c, ca_{c,t}, loc_{c,t}, Pr_{c,t}, Paw_{c,t}, PAPM_{c,t}, d, ap_{c,t}, csgaf_c)$

cag A consumer agent group $cag \in CAG$ is defined as an 11-tupel
$cag = (CGA_{cag}, \mathscr{SD}_{cag}, \mathscr{CP}_{cag}, \mathscr{PAW}_{cag}, \mathfrak{ppsm}_{cag}, d, NDS_{cag}, CS_{cag}, \mathscr{IFPC}_{cag}, \mathscr{FPAD}_{cag}, ia_c$

$copm$ A *copm* is a 4-tupel $(c, \hat{c}, pa, p) = copm$, as a message stemming from sender c to receiver $\hat{c}$ about product attribute pa of product p, changing the perception of the receiving consumer by adding a perception of a strength corresponding to the edge weight between the two consumers, making the receiver aware of a product attribute if she isn't already

d A specific decision procedure $d \in D$

e A directed edge in the social network $e = (r, o, m) \in E$ from node r to node o regarding medium m

es An event scheduler $es \in ES$ that can be used in the process model

fp A fixed product used in the simulation ($fp \in FP$):
$fp(t) = (PA_{fp}(t), pga_{fp}, pas_{fp,t}, \mathscr{PL}_{fp}$

m A medium in which communication along an edge will take place

n A need within the simulation: $n \in N$

p A product $P \ni p = (PA_p(t), pga_p, pas_{p,t}, \mathscr{PL}_p)$ is defined through its product attribute set $PA_p \subset PA$, its associated product group ($pga : \mathfrak{pgam}(p) = pga$), its product activation status $pas_{p,t} = \mathfrak{apm}$ and its product lifetime distribution $\mathscr{PL}_p$

pa Product attribute $pa \in PA$ as the value $pav_{pa,t} = \mathfrak{pavm}(pa, t)$ it takes along a quality dimension, its mutability $pam_{pa} = \mathfrak{pamm}(pa)$ and observability $pao_{pa} = \mathfrak{paom}(pa)$ over time:
$pa(t) = (pav_{pa,t}, pam_{pa}, pao_{pa})$

pai	A product attribute information $pai \in PAI$ is an information that relates to a product attribute (see PAI for elaboration)
pg	A product group $pg \in PG$ is described as an 8-tupel $pg = (PGA_{pg}, PPG_{pg}, EPG_{pg}, FP_{pg}, pgn_{pg}, SP_{pg}, odp, \mathscr{PL}_{pg})$
pga	A product group attribute $pga \in PGA$
pos	A point of sale $pos \in POS$ is a 5-tupel $pos_t = (Av_{pos}(t), PPr_{pos}(t), loc_{pos}, ia_{pos}, PuPS_{pos})$
ps	A perception scheme $ps \in PS$ used in the simulation, describing the dynamics of a perceived product attribute value map $\mathfrak{ppavm}$
r, o	A node in the social network $r, o \in AN$
se	A scripted event $se \in SE$
SP_{pg}	The standard product for the product group pg
v	A value $v \in V$
coa	A company agent within the simulation, i.e. $coa \in COA$. A coa is parameterized as a 5-tupel $coa(t) = (PP_{coa}(t), PQMS_{coa}, MDS_{coa}, AS_{coa}, ia_{coa})$.
ewds	An edge weight dynamics scheme $ewds \in EWDS$ to be used in an EWS
ewis	An edge weight initialization scheme $ewis \in EWIS$ to be used in an EWS
tms	A topology manipulation scheme $tms \in TMS$ to be used in a social network
$\mathfrak{M}$	Model of the system simulated
$\mathfrak{S}$	The spatial model, describing spatial aspects of the model $\mathfrak{M}$
$ca_{c,t}$	The state of the consumer attributes $ca_{c,t}$ of consumer agent c at time t, with the vector of consumer attributes being defined by the corresponding consumer agent group cag_c, in order to express consumer agent heterogeneity
cag_c	The Consumer Agent Group Association $cag_c = \mathfrak{cagm}(c)$, indicating which consumer agent group c is a member of ($\mathfrak{cagm} : C \rightarrow CAG$)
G	Social graph: The graph representing connections in the social network $G = (AN, E)$
$loc_{c,t}$	The coordinates of consumer agent c at time t within the spatial model $\mathfrak{S}$
odp_{pg}	The overwrite decision process of product group pg determines which decision process is chosen for decision overwrite, and parameterized through the respective map ($odp_{pg} = \mathfrak{odpm}(pg)$), if defined for this product group

$PA_p(t)$ Product attributes of product p at time $t \in T$. The state of a product attribute at time t is described by the 3-tupel $pa(t) = (\mathfrak{pav}(pa,t), \mathfrak{pam}(pa), \mathfrak{pao}(pa))$

pam_{pa} The product attribute mutability of pa: $pam_{pa} = \mathfrak{pamm}(pa)$, describing whether the value of the product attribute is allowed to change

pao_{pa} The product attribute observability of pa: $pao_{pa} = \mathfrak{paom}(pa)$, specifying how observable (as a number on the unit interval) the product attribute is

$pas_{p,t}$ A products activation status $pas_{p,t} = \mathfrak{apm}(p,t)$, indicating whether the product is 'active' in the market at a given time, where active means they are introduced to the market already and are not (yet) discontinued: $\mathfrak{apm} : P \times T \to \mathbb{B}$

$pav_{pa,t}$ the product attribute value of pa at time t: $pav_{pa,t} = \mathfrak{pavm}(pa,t)$, being the numerical magnitude of the quality

$pgam_{pga}$ The (binary) product group attribute mutability value, describing the probability whether the product group attribute pga is mutable

$pgao_{pga}$ The product group attribute observability value, describing the observability of product attribute pga

PM The process model as a pair $PM = (es, ars)$ of the event scheduler $es \in ES$ and the adoption replacement scheme $ars \in ARS$

poa The policy agent within the simulation, representing all endogeneous and exogeneous policy aspects; It is formalized as 5-tupel $poa = (PPS_{poa}, CPS_{poa}, RPS_{poa}, MES_{poa}, ia_{poa})$

$PP_{coa}(t)$ The product portfolio of agent coa at time t, $PP_{coa}(t)$, describes what products the company agent manages (which should partition the set of products in disjoint sets, i.e. $\bigcup\limits_{coa \in COA} \bigcup\limits_{t \in T} \bigcup\limits_{p \in PP_{coa}(t)} p = P = \bigcup\limits_{t \in T} P_t$ and $PP_{coa}(t) \cap PP_{\hat{coa}}(t) = \emptyset \; \forall \text{coa} \neq \hat{coa}$)

$PuPS_{pos}$ The product purchase process used by point-of-sale pos that consumer agents engage in when purchasing at pos

SN The social network comprising of the social graph G, the edge weight scheme ews and the topology manipulation scheme tms, i.e. $SN = (G, ews, tms)$

$\mathscr{CP}_{cag}$ The preference distribution set $\mathscr{CP}_{cag}$ associated with cag ($\mathscr{CP}_{cag} = \bigcup\limits_{v \in V} \mathscr{CP}^v_{cag}$, with $\mathscr{CP}^v_{cag}$ being the consumer agent preference distribution for value v for the consumer group cag; used to determine the initial preferences of an agent $c \in \mathfrak{cagm}^{-1}(cag)$)

$\mathscr{PPAV}_{cag}$ The set of perceived product group attribute value distributions associated with consumer agent group cag ($\mathscr{PPAV}_{cag} = \bigcup\limits_{pga \in PGA} \mathscr{PPAV}^{pga}_{cag}$)

AN The set of nodes in the social network. Each node in the network $r, o \in AN$ is part of the description of the agent

$ap_{c,t}$ The set of adopted product $ap_{c,t} = \mathfrak{pam}(c, t)$, indicating which products are adopted by $c \in C$ at time $t \in T$:
$\mathfrak{pam} : C \times T \to P^{ad}_{c,t} \subset Pot(P)$, 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 a set $ap_{c,t} \in P^{ad}_{c,t}$, as the potential product combinations to adopt (subset of all products P)

ARS The set of adoption replacement schemes that can be used for the model.

C The set of consumer agents

CAG The CAG is the index set of the consumer agent groups

$COPM$ The set of consumer perception messages, with $copm \in COPM = \bigcup_{cag \in CAG} COPM_{cag}$

$COPM_{cag}$ The set of consumer perception messages of consumer group $cag \in CAG$

D The set of decision procedures in the model

E The set of edges within the social graph $G = (AN, E) = (AN, E(t))$

EPG_{pg} The set of exclusive product groups for the product group pg, $(EPG_{pg} \subset PG)$

ES The set of event scheduler that can be used for the model.

FP The set of fixed products used in the simulation: $FP = \bigcup_{t \in T} FP_t$

FP_{pg} The set of fixed products of product group pg

FP_t The set of fixed products at time $t \in T$

IA The set of information agents, i.e. agents that can convey information and require an information authority ia_a for an information agent $a \in IA = C \cup POS \cup CA \cup poa$

M The set of media in which the communication between agents in the social network takes place

N The set of needs used in the simulation

P The set of products used in the simulation (regardless of their product group and simulation time)

PA The set of product attributes used in the simulation. Since new products can enter the simulation and existing products can be discontinued, this set includes all product attributes that exist at some point within the simulation ($PA = \bigcup_{p \in P} \bigcup_{t \in T} PA_p(t)$)

PAI The product attribute information $PAI \subset I$ is the subset of information that relates to a product attribute information as a subset of all information (and implicitly specifies the domain for which $\mathfrak{ipam}$ is well-defined

PG The set of product groups in the simulation

PGA The set of product group attributes

PGA_{pg} The set of product group attributes associated with product group pg: $PGA_{pg} \subset PGA$

pgn_{pg} The product group need set $pgn_{pg} \subset N$ of product group pg specifies the needs $n \in N$ that the product group pg satisfies. It is parameterized through the product group need map $\mathfrak{pgnm}$ through $pgn_{pg} = \mathfrak{pgnm}(pg)$

POS The set of POS

PPG_{pg} The set of prerequisite product groups for the product group pg, $(PPG_{pg} \subset PG)$

PS The set of perception schemes used within the simulation, i.e. $PS = \bigcup ps$

SE The set of scripted events, i.e. market introduction events and product discontinuation events

V The set of values used in the simulation

COA The set of company agents within the simulation.

EWDS The set of Edge weight dynamics schemes contains all the edge weight dynamics schemes $ewds \in EWDS$ that can be used by the edge weight scheme

EWIS The set of Edge weight initialization schemes contains all the edge weight initialization schemes $ewis \in EWIS$ that can be used by the edge weight scheme

EWS The set of Edge weight schemes contains all the edge weight schemes $ews \in EWS$ to be employed in the social network

TMS The set of topology manipulation schemes containing all the topology manipulation schemes $tms \in TMS$ to be employed in the social network

Av_{pos} The product availability vector $Av_{pos}(t)$ of pos $(Av_{pos}(t) = (av_p^{pos,t}) \in \mathbb{B}^{\mathcal{N}_P}$, indicating whether the respective products p are available at the time t)

CGA_{cag} 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 (i.e. the entries are related 1-to-1 with each entry of the consumer group attribute vector corresponding to the consumer agent attribute derived from it and vice versa)

EPG A vector of excluding product group where each entry (corresponding to product group pg is a set of product groups $EPG_{pg} \subset PG$ that are exclusive for pg)

$PAPM_{c,t}$ The product attribute perception vector of consumer agent c, indicating which perception c has of the value of a certain product attribute pa at time t as a numerical value ($PAPM_{c,t} = (papm_{pa}^{c,t}) \in \mathbb{R}_{\geq 0}^{\mathcal{N}_{PA}^{t}}$, with the entry corresponding to pa being defined through $\mathfrak{ppavm}(c, pa, t)$)

$Paw_{c,t}$ The product awareness vector for time point t and actor c, indicating what products c aware of ($Paw_{c,t} = (paw_{p}^{c,t}) \in \mathbb{B}^{\mathcal{N}_P}$, $Paw_{c,t} = (paw_{p}^{c,t}) \in \mathbb{B}^{\mathcal{N}_P}$, with $paw_{p}^{c,t} =$ true meaning that c is aware of p at time t and $paw_{p}^{c,t} =$ false meaning it isn't)

PPG A vector of prerequisite product group where each entry (corresponding to product group pg is a set of product groups $PPG_{pg} \subset PG$ that are prerequisite for pg)

PPr_{pos} The POS price vector of pos with $PPr_{pos}(t) = (PPr_{p}^{pos,t}) \in \mathbb{R}_{\geq 0}^{\mathcal{N}_P}$ at time t

$Pr_{t,c}$ Preference vector: Describes the preference of fixed consumer agents c for value v at time t ($Pr_c = (pr_v^c) \in \mathbb{R}^{\mathcal{N}_v}, (pr_v^c)(t) = \mathfrak{pm}_c(v, t)$)

SP A vector of standard products where each entry (corresponding to product group pg is a a product p $SP_{pg} \in P$ that is the standard product for pg)