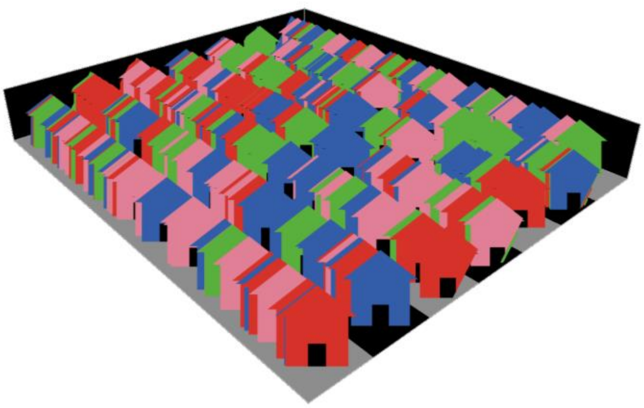
European Commission

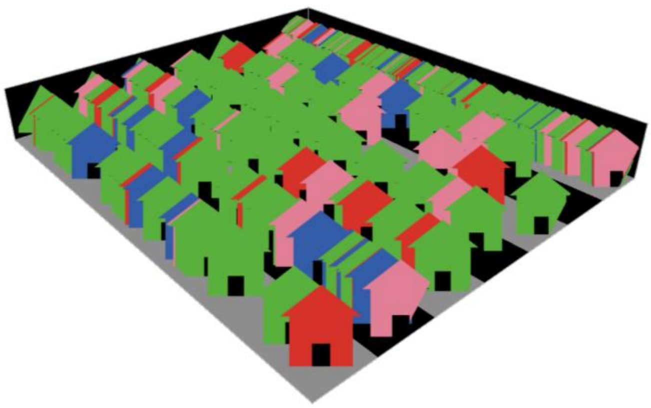
JRC TECHNICAL REPORTS

SIMP - Subjective Individual Model of Prosumer

Vasiljevska, J., Mengolini, A.,
Nikolic, I.

2018





Joint Research Centre

EUR 29514 EN

EU Science Hub
https://ec.europa.eu/jrc

JRC113203

EUR 29514 EN

PDF ISBN 978-92-79-98285-9 ISSN 1831-9424 doi:10.2760/786611

Luxembourg: Publications Office of the European Union, 2018

How to cite this report: Vasiljevska, J., Mengolini, A., Nikolic, I., *Subjective individual model of prosumer*, EUR 29514 EN, Publications Office of the European Union, Luxembourg, 2018, ISBN 978-92-79-98285-9, doi:10.2760/786611, JRC113203

Contents

Authors

Julija Vasiljevska, European Commission, Joint Research Centre

Anna Mengolini, European Commission, Joint Research Centre

Igor Nikolic, Energy and Industry Group, Faculty of Technology, Policy and Management, Delft University of Technology

Abstract

The energy consumer is at the centre of the European Union's energy policies. Consumer's active participation is considered as a prerequisite for managing the energy transition successfully and in a cost-effective way. The recent measures proposed by the European Commission with the 'Clean Energy for all Europeans' rely on smart grid technologies, solutions and concepts to accelerate, transform and consolidate the EU clean energy transition.

In this context, the aim of this report is to present an agent based model of the electricity consumer (SIMP – Subjective Individual Model of Prosumer). The model can be used as a tool to better understand the impact that innovative energy services, enabled by smart grid technologies, may have on the electricity consumers and the society at large. Furthermore, the model can be used as a tool to gain insight into diffusion patterns of energy services (in this report represented by electricity contracts) and associated switching rates. As such, it contributes to the understanding of what fosters and what hinders an effective deployment of innovative energy services.

1 Introduction

The objective of the agent-based model SIMP – Subjective Individual Model of Prosumer – is to explore the diffusion of energy services, enabled by smart metering technologies, among a population of interacting electricity prosumers and to evaluate the impact of such diffusion on individual and societal performance indicators. Agent Based Modelling (ABM) is increasingly being considered as a suitable tool to address the complexity of socio-technical systems that are characterized by a strong interaction between the human and the technical system (Gilbert, 2008) (van Dam et al., 2013). Therefore, ABM represents an appropriate tool to describe and simulate the complexity of the consumer role in the emerging energy systems.

The model architecture, model parameters and model simulation have been presented in "An agent-based model of electricity consumer: smart metering policy implications in Europe" (Vasiljevska et al., 2017) and in "Prosumer behaviour in emerging electricity systems" (Mengolini, 2017). The aim of this report is to present the model structure and code in more details and as such, it should be seen as supportive documentation to the documents mentioned above. SIMP is implemented in Netlogo (Wilensky, 1999).

The report is structured as follows: chapter 2 describes the model pseudocode by making a clear reference to the source code. Chapter 3 sheds some light on the future developments of the model, where the authors present an integrated framework under which the current model could be further extended. The source code of the model is included in the annex.

2 SIMP – Subjective Individual Model of Prosumer

The model consists of a number of agents, representing households and of a portfolio of electricity contracts, representing different energy services offered by an electricity retailer.

Consumers are modelled as household agents having dynamic preferences on the types of electricity contracts offered by the retailer. Development of these preferences depends on consumers' personal values, memory and attitudes, as well as the degree of interaction in a social network structure.

2.1 What is it

The model aims to provide better understanding of the impact that smart metering systems may have on the electricity consumers and the society at large, and in particular on the deployment of thereby enabled energy services. Therefore, the model can be used as a tool to gain insight into diffusion patterns of energy services (represented by an electricity contract) and associated switching rate among contracts.

2.2 How it works

Electricity consumers are modelled as household agents having dynamic preferences on types of electricity contracts offered by the retailer. The agents interact with the electricity network and the retailer through electricity contracts and develop preferences on different contract types, depending on their *personal values*, *memory* and *attitudes*, as well as the degree of *interaction* in a *social network* structure. In this regard, the agents and the electricity network represent a socio-technical system whose behaviour may be influenced by the experience of other consumers, as well as the behaviour of other actors, such as for example, retailers and national/local authorities through market dynamics and national policies. While such policies and institutions can be influenced and shaped by agent's behaviour and evolve over time, for the purpose of this model they stay exogenous and fixed.

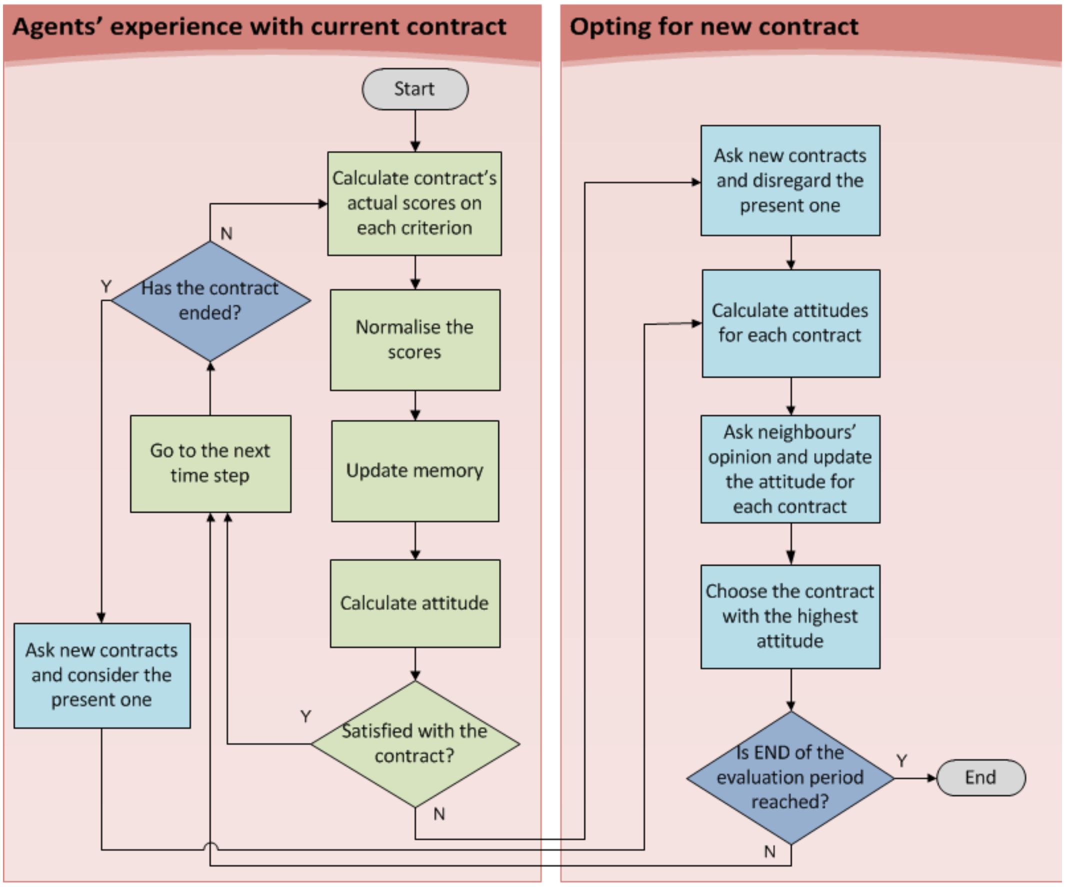
The decision making process is modelled as a multi-criteria decision making problem and it consists of two sub-models, as presented in Figure 1.

Sub-model I: Decision making on switching to different contract

Each time step an agent gets experience with a certain contract (NetLogo procedure *calc-experienced-outcomes*, **Figure 2**), which is modelled as a random value drawn from a predefined contract specific set (Table 3). This experience is then normalised and put in the agent's memory, and it is updated with every new experience (NEtLogo procedure *normalise-experienced-outcomes,* **Figure 2**). Based on the updated score and the agent's specific weighting factors (Table 2, *agent-specific variables*) - which reflect the relative importance an agent gives to certain criterion: financial savings, comfort change, CO_2 savings and social welfare - the agent develops an overall attitude towards a certain contract (NetLogo procedure *calc-attitudes,* **Figure 2**).

Ultimately, based on this attitude and the agent-specific attitude threshold (Table 2, *simulation parameters*), the agent decides to switch to a different contract or stay with the current one (NetLogo procedure *am-i-satisfied*, **Figure 2**).

Figure 1. Agents' decision making process



Source: own elaboration

Figure 2 provides a screenshot of the main NetLogo procedures related with the sub-model I.

Figure 2. Screenshot of the NetLogo code file (sub-model I)

```
to calc-experienced-outcomes
  let pi-nr 0
  ;iterate over the perfromance indicators, i.e. outcomes
  while [ pi-nr < nr-of-perf-ind ] [
    ;calculate the experienced outcome
    let min-outcome matrix:get communicated-outcomes current-contract-nr (pi-nr * 2)
    let max-outcome matrix:get communicated-outcomes current-contract-nr (pi-nr * 2 + 1)
    let outcome min-outcome + random-float (max-outcome - min-outcome)
    ;put the outcome in the list for the current time step
    set experienced-outcomes replace-item pi-nr experienced-outcomes outcome
    ;add the outcome to the total outcomes
    set total-outcomes replace-item pi-nr total-outcomes ( ( item pi-nr total-outcomes ) + outcome )
    set pi-nr pi-nr + 1]
end
to normalise-experienced-outcomes
  let loop-pi-nr 0
  let normalized-experienced-outcomes n-values nr-of-perf-ind [-100]
  foreach experienced-outcomes [
    let experienced-outcome ?
    let normalized-experienced-outcome normalize loop-pi-nr experienced-outcome
    set normalized-experienced-outcomes replace-item loop-pi-nr normalized-experienced-outcomes
    normalized-experienced-outcome
    set loop-pi-nr loop-pi-nr + 1]
    ;put the experience with the current contract into the memory
  matrix:set-row normalized-experienced-outcomes-memory current-contract-nr
  normalized-experienced-outcomes
end
to calc-attitudes
  ;take the normalized outcomes out of the memory
  let normalized-experienced-outcomes matrix:get-row normalized-experienced-outcomes-memory
  current-contract-nr
  ;calculate agent's attitude towards the current contract and store it
  let attitudes-current-contract calc-contract-attitudes normalized-experienced-outcomes
  ;put the attitudes into the attitudes matrix
  matrix:set-row attitudes current-contract-nr attitudes-current-contract
  ;calculate the general attitudes
  let general-attitude sum (matrix:get-row attitudes current-contract-nr)
  ;insert the general attitude of the current contract into the list of general attitudes
  set general-attitudes replace-item current-contract-nr general-attitudes general-attitude
end
to-report am-i-satisfied
  let general-attitude item current-contract-nr general-attitudes
  ; compare the average attitude to the threshold
  report general-attitude >= threshold-attitude
end
```

Sub-model II: Choosing contract from portfolio of contracts

The contract choice is modelled as a multi-criteria decision making problem and follows the same structure for all agents. The agent receives a portfolio of contracts and based on her perception of technological risks associated with each contract (Table 2*, contract-specific variables*) and agent's specific *techno-tolerance threshold* (Table 2, *agent-specific variables*), the agent evaluates only contracts with perceived technological risks above her *techno-tolerance threshold*, whereas the rest of the contracts are discarded. The evaluation process includes agent's personal attitudes (NetLogo procedure *evaluate-contract*, **Figure 3**) as well as attitudes of peers' agents, *i.e.* social influence (NetLogo procedure *ask-neighbours-opinion*, **Figure 3**). Finally, the best scoring contract is the one to be adopted (NetLogo procedure *choose-contract*, **Figure 3**).

Figure 3 provides a screenshot of the main NetLogo procedures related with the sub-model II.

Figure 3. Screenshot of the NetLogo code file (sub-model II)

```
to evaluate-contract [ contract-nr ]
  ;get the normalized average communicated outcomes
  let temp-communicated-outcomes matrix:get-row normalized-avg-communicated-outcomes contract-nr
  ;calculate attitudes
  let attitudes-temp-contract calc-contract-attitudes temp-communicated-outcomes
  ;put these attitudes into the prosumers-own matrix of expected attitudes
  matrix:set-row own-expected-attitudes contract-nr attitudes-temp-contract
end


to ask-neighbours-opinions [ contract-nr ]
  ;sum the attitudes of different neighbours
  let summed-neighbours-attitude 0
  let nr-of-neighbours-with-experience 0
  ;initialise provisional value for the number of contacts
  let nr-of-contacts -100
  let loop-nr-of-contacts 0
  while [loop-nr-of-contacts < nr-of-interactions] [
    ifelse random-float 1 > heterogeneity [
      ask n-of 1 prosumers with [[archetype] of self = [archetype] of myself ] [
        ;if the prosumer has experience
        if item contract-nr general-attitudes > -100 [
          set nr-of-neighbours-with-experience nr-of-neighbours-with-experience + 1
          ;get the neighbours' attitude for this contract
          let neighbours-attitude item contract-nr general-attitudes
          ;add the neighbours'attitude to the summed attitudes
          set summed-neighbours-attitude summed-neighbours-attitude + neighbours-attitude
        ]]]
      ;introduce heterogeneity, i.e. ask opinion from agents belonging to archetype different
      ;than the agent's own
      [ask n-of 1 prosumers with [[archetype] of self != [archetype] of myself ] [
        if item contract-nr general-attitudes > -100 [
        ;if the prosumer has experience
          set nr-of-neighbours-with-experience nr-of-neighbours-with-experience + 1
          ;get the neighbours' attitude for this contract
          let neighbours-attitude item contract-nr general-attitudes
          ;add the neighbours'attitude to the summed attitudes
          set summed-neighbours-attitude summed-neighbours-attitude + neighbours-attitude
        ]]]
    set loop-nr-of-contacts loop-nr-of-contacts + 1] ; end of while-loop
  ;initialise provisional value
  let influenced-expected-general-attitude -100
  ;if there is at least 1 neighbour with experience
  ifelse nr-of-neighbours-with-experience > 0 [
    ;calculate the average neighbours' general attitude for the respective contract
    let avg-neighbours-attitude summed-neighbours-attitude / nr-of-neighbours-with-experience
    ;create a temporary variable with the agent's own expected attitude for the respective contract
    let own-expected-general-attitude item contract-nr own-expected-general-attitudes
    set influenced-expected-general-attitude own-expected-general-attitude +
    (avg-neighbours-attitude - own-expected-general-attitude) * susceptibility]
    ;if there are no neighbours with experience, take agent's own expected attitude
  [  set influenced-expected-general-attitude item contract-nr own-expected-general-attitudes]
  set influenced-expected-general-attitudes replace-item contract-nr influenced-expected-general-attitudes
  influenced-expected-general-attitude
end


to choose-contract
   let old-contract-nr current-contract-nr
   ;choose the contract with the best general attitude from the available contracts
   let max-general-attitude max influenced-expected-general-attitudes
   set current-contract-nr position max-general-attitude influenced-expected-general-attitudes
   ;update the visual representation considering the new chosen contract
   set xcor current-contract-nr
   if old-contract-nr != current-contract-nr [
   set changed-contract? true]
end
```

2.3 How to use it

The key questions the model tries to answer are: 1) how smart metering technologies, and thereby enabled services can be promoted under different policy settings and effectively adopted by the consumers and 2) how this technological diffusion affects individual and societal performance indicators. The outcomes are, however, not meant as predictions, but rather an exploration of the mechanisms at play.

Below we provide a description of the variables (also summarised in Table 2) and the main outputs, as observed in the NETlogo interface tab (Figure 4).

Figure 4. Screen shot of the SIMP GUI (at the beginning and at the end of the simulation)

Policy: The model is intended to explore possible diffusion rates of smart metering under different policy interventions and, as a result, it provides an insight into diffusion patterns of energy services and associated consumers' switching rates. In this context, and in line with the current EU developments in the area of smart metering, three policy interventions have been subject to experimental setup and data analysis in the model (presented as slider in the GUI, Figure 4):

- Policy 0 – mandatory installation of smart metering systems, which mandates the Distribution System Operator (DSO) to install a smart meter at the consumer's premises and the consumer is required to accept the meter.
- Policy 1[1] – voluntary installation of smart metering systems where the consumer can refuse the smart meter ("opt-out option") or turn off the option to remotely exchange consumption data with the supplier or any third party and to be remotely disconnected ("administrative-off" option).
- Policy 3 – voluntary installation of smart metering systems encouraged by an environmental campaign, and thereby promoting the environmental benefits of smart metering enabled services.

Initial contract distribution: *consumers-contract-A, consumers-contract-B, consumers-contract-C, consumers-contract-D, consumers-contract-E, consumers-contract-F, consumers-contract-G* represent the set of contracts available to the consumer at the beginning of the simulation (depending on the policy, see Table 1). Each contract is characterised with *contract duration*, *individual and societal indicators* (financial and CO_2 savings, comfort change and social welfare) and *perceived technological risk* associated with the smart metering system. All contracts are assumed to be initially equally distributed among the agents. Additionally, further experiments with different initial contract distributions have been performed (Table 4). Table 1 illustrates the types of contracts available in each policy.

Table 1. Available contract types per policy

	Contract						
Policy	**A**	**B**	**C**	**D**	**E**	**F**	**G**
1	No	Yes	Yes	No	Yes	Yes	Yes
2	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Source: own elaboration

Contract-duration: we assume that each contract has a *duration* of 12 months (minimum contract duration observed in most EU member states). Alternatively, the model is tested under the assumption of indefinite contract duration so as to analyse the effect of "lock-in" periods, during which the consumer would need to pay a penalty in case she ends up the contract prematurely.

Share-ego, share-hedo, share-bio, share-alt define consumers' archetype. Consumer agents are characterised by *weighting factors* (or *weights*), which reflect the relative importance an agent gives to a certain criterion (hereinafter called performance indicator): financial savings, comfort change, CO_2 savings and social welfare. The *weights* are randomly assigned to each agent, following a uniform distribution [0; 1] and they are

[1] the case of the Dutch smart metering roll-out

normalised so that the sum of the *weights* assigned to each agent equals to 1. The highest *weighting factor* determines the archetype the agent belongs to. Initially, we assume an equal distribution of each archetype.

Threshold-attitude is used to measure the agent's satisfaction with a certain contract. If the overall attitude is lower than the *threshold attitude*, the agent decides to change contract.

Heterogeneity represents the number of peers belonging to a different archetype than agent's own. An agent may decide to adopt a certain contract based on the experience other agents in the social network have with the same contract and the value the agent gives to the opinion of those other agents (i.e. *susceptibility*).

Susceptibility measures the importance the agent gives to the opinion of her peers regarding a certain type of contract. It is drawn from a uniform distribution between *min susceptibility* and *max susceptibility* (presented as slider in the GUI, Figure 4).

Nr-of-interactions: interaction occurs through social influence among household agents interconnected in the social network. In such network, agents communicate directly with a randomly chosen number of agents and the interactions depend on the heterogeneity level and susceptibility.

Table 2 describes the model parameters and variables used in the simulation. Empirical values were not available for most of the parameters and as a result, synthetic data were used, based on expert judgment and the same were extensively varied. Nevertheless, wherever a source is given, the parameter value is empirically based.

Table 2. Model parameters and variables used in the simulation

Simulation parameters

Variable name	Brief description	Value	Source
nr-of-interactions	Number of peers each agent communicates with	5	-
policy	The policy determines what contracts are available, and whether an environmental campaign is introduced at t=40 months	mandatory, voluntary, environmental	
heterogeneity	Number (in terms of %) of peers belonging to different archetype than the agent's own.	0.5	-
Initial contract distribution	Contract distribution among agents at the beginning of the simulation	Equal distribution or all agents have the least technologically advanced contract (contract A or B, depending on the policy)	-
threshold-attitude	Used to measure the agent satisfaction with the certain contract. If the overall attitude is lower than the threshold-attitude, agent decides to change contract.	0.5	-
contract-duration	Contract time duration	12 months or indefinite	-

Simulation variables depending on the parameterization

Variable name	Brief description	Value	Source
N	Number of household agents, depending on the policy	200 (policy 0) or 280 (policy 1 and 2)	-
Available contracts	The contracts that are available to the consumers, depending on the policy	{B,C,E,F,G} or {A,B,C,D,E,F,G}	-
Environmental campaign	Determines whether or not an environmental campaign is introduced in month 40 and it depends on the policy	yes or no	-

Agent-specific variables

Variable name	Brief description	Value	Source
Weighting factors: w_e, w_h, w_b, w_a	Measure of relative importance agent gives to certain criterion (egoistic, hedonic, biospheric, altruistic)	Chosen from uniform distribution [0,1]	-
Susceptibility w_{SN}	Measure of the importance agent gives to the opinion of her social network peers	0.5	-
techno-tolerance threshold	Contract acceptance level due to perceived risks associated with smart metering technology	[1,11]	-

Contract-specific variables

a_{emin}, a_{emax}	communicated range for financial savings	See Table 3	Eurostat statistics explained
a_{hmin}, a_{hmax}	communicated range for comfort change	See Table 3	B. Boardman et al.
a_{bmin}, a_{bmax}	communicated range for CO_2 savings	See Table 3	Covenant of mayors, 2010
a_{amin}, a_{amax}	communicated range for social welfare	See Table 3	S. Darby et al., 2012
techno-risks	perceived technological risks	[1, 6]	-

Source: own elaboration

Table 3. Communicated outcomes per contract type

Contract	Financial savings [€]	Comfort change [%]	CO_2 emissions savings [t]	Social welfare [%]
A	[0; 0.1]	[0; 0.1]	[0; 0.5]	[10; 20]
B	[0.54; 0.66]	[0; 0.1]	[1; 2]	[10; 20]
C	[0.54; 0.66]	[0; 0.1]	[1; 2]	[10; 20]
D	[1.08; 1.32]	[-6; -4]	[2; 4]	[20; 30]
E	[2.2; 2.6]	[-12; -8]	[5; 7]	[20; 30]
F	[3.2; 4]	[-12; -8]	[8; 10]	[30; 40]
G	[5; 7]	[-17; -13]	[12; 16]	[40; 50]

Source: own elaboration

Average-tolerance-satisfaction: average value of the *techno-tolerance-satisfaction*, described as the difference between the agent specific *techno-tolerance-threshold* (Table 2) and the technological risks (*techno-risks*, Table 2) associated with the adopted contract. The *techno-tolerance-threshold* is initialized at the beginning of the simulation as a random number drawn from a predefined set and the *techno-risk* has predefined value for each contract, based on the smart meter functionalities enabled for the specific contract, and it does not change during the model simulation (Table 2, *contract-specific variables*).

Average-attitude-satisfaction: average value of the *attitude-satisfaction* described as a difference between the agent's general attitude regarding a certain contract and the agent specific attitude threshold (*threshold-attitude*, Table 2).

Contract adoption: plots the total distribution of all contracts at each time step, during the whole simulation period.

Contract adoption ego: plots the total distribution of contracts among the egoistic archetype of agents at each time step, during the whole simulation period.

Contract adoption hedo: plots the total distribution of contracts among the hedonic archetype of agents at each time step, during the whole simulation period.

Contract adoption bio: plots the total distribution of contracts among the biospheric archetype of agents at each time step, during the whole simulation period.

Contract adoption alt: plots the total distribution of contracts among the altruistic archetype of agents at each time step, during the whole simulation period.

Experimental set-up

As previously mentioned, we are interested to understand the impact of the agents' switching behaviour on system level outcomes, such as *adoption of contract types*, *average financial savings*, *average CO_2 savings*, *average comfort change* and *average social welfare.* The parameter values vary between runs due to the stochastics used during agents' initialization and model execution. Therefore, to be able to arrive at realistic assessment of patterns observed in the simulated system evolution, we need to do a statistical analysis of the results of many runs. To this end, we build an experimental set-up relative to the following variables: *heterogeneity*, *policy*, *initial contract distribution* and *contract duration*. Heterogeneity, however, as modelled in SIMP does not prove to have impact on the average contract distribution. This is due to the fact that *egoistic*, *biospheric* and *altruistic* agents have objectives which pull in the same direction, in terms of contract type preference, i.e. agents who belong to these three archetypes will behave similarly, whereas only hedonic agents will act differently. Therefore, the experimental set-up was build relative to the initial contract distribution and contract duration (Table 4) and each experiments was tested for each policy separately. The parameterisation for the simulation experiments is given in Table 2. The programming language R was used for the data analysis and the results are presented and extensively discussed in (Vasiljevska, Douw, Mengolini, & Nikolic, 2017).

Table 4. Experimental set-up

Scenario	Initial contract distribution	Contract duration
1	All agents have the least technologically advanced contract (B in the mandatory policy and A in the voluntary and environmental policy)	12 months
2	Equal contract distribution	12 months
3	All agents have the least technologically advanced contract (B in the mandatory policy and A in the voluntary and environmental policy)	Indefinite
4	Equal contract distribution	Indefinite

Source: own elaboration

An average contract distribution is observed at each time step to understand the reasoning behind the contract adoption patterns and influencing factors (incentives, social influence, etc.). Finally, average technology satisfaction and average attitude satisfaction have been analysed in each policy and among scenarios to understand the link between system level performance (in terms of financial savings, CO_2 emissions reduction, etc.) and the agents' satisfaction level. A distinct pattern of distribution of contracts, system level outcomes and satisfaction level emerge from the analyses presented in (Vasiljevska, Douw, Mengolini, & Nikolic, 2017).

2.4 Things to notice

High switching rate among different contracts can be observed in the NETlogo interface tab (Figure 4): each household represents a consumer and it is assigned a colour defining the archetype she belongs to. Each column inside the patch monitor of the NetLogo interface tab (Figure 4) represents a contract type (NB: 5 contract types are available in policy 0 and 7 contract types are available in policy 1 and 2).

The overall time distribution of contract adoption can be monitored in the graph *contract adoption* of the interface tab. Additionally, the time distribution of different contract adoptions per different consumer's archetype can be observed in the graphs *contract adoption ego*, *contract adoption hedo*, *contract adoption bio* and *contract adoption alt*.

We can see that more technologically advanced contracts, such as contract F and G are highly adopted among all consumers' archetypes. This is caused by the fact that these are the best-scoring contracts for the 3 indicators (financial savings, CO_2 savings and social welfare). Less technologically advanced contracts, such as A, B and C (in policy 2) are highly adopted among the hedonic type of consumers.

The average attitude satisfaction is mostly close to zero or sometimes negative, which indicates high average switching rate. This can be associated with the experience agents get each time step, which is modelled as exogenous variable randomly drawn from a predefined set (specific for each contract). Also, the techno-tolerance threshold is exogenous, fixed at the initialization of the model, which prevents the agents to consider more "technologically risky" contracts that would yield better outcome (in terms of energy savings, CO_2 reduction, etc.). This can be the reason why we observe high adoption rate of less technologically advanced contracts (e.g. contract A and B) also among the biospheric type of consumers.

2.5 Things to try

When moving the policy slider in Figure 4 , we observe the following:

- More technologically advanced contracts, such as contract F and G remain highly adopted in all policies.
- There is a major difference in the adoption level of contract A and contract D when comparing the mandatory policy (policy 0) on the one hand, and the voluntary and environmental policy on the other hand (policy 1 and 2, respectively). This difference is caused by the fact that contract A and contract D are not available in the mandatory policy.
- There are no significant differences in the adoption level of the contract types between the voluntary (policy 1) and environmental policy (policy 2) even though one would expect that the environmental policy and the associated increase of biospheric consumers would yield a higher share of more advanced contracts. This can be explained by the fact that the techno-tolerance threshold, as currently modelled, does not vary by archetype. As a result, increase in the number of biospheric consumers does not necessarily lead to increased adoption of more technologically advanced contracts, as biospheric consumers can still be highly concerned with the technological risks associated with more advanced contracts.
- Granting the consumers opt-out and "administrative-off" option for smart metering system (in policy 1) results in increased number of consumers opting for a less technologically advanced contract (i.e., contract A or D). Also, we see that granting consumers more options (as a way to tackle technology related concerns), does not necessarily lead to higher energy and CO_2 savings and ultimately higher consumers' satisfaction. This effect remains strong even under the environmental policy (policy 2) where despite significant number of agents becoming more environmentally concerned, the adoption of contracts that do not require data sharing with DSOs (e.g. contract A and D) still remains significant.
- Technologically advanced contracts that yield higher benefits may also be subject to technology associated concerns, as perceived by the consumers. In this respect, the average techno-tolerance satisfaction appears to be the lowest in the environmental policy, where consumers opt for more technologically advanced contracts.
- Moving the policy slider and varying the contract duration along with the initial contract distribution do not seem to significantly affect the average total-attitude satisfaction. This can be associated with the high diversity of the agents' population (in terms of agents' archetype).
- Moving the heterogeneity slider in Figure 4, does not have significant impact on the average contract distribution. This is due to the fact that egoistic, biospheric and altruistic agents have objectives which pull in the same direction, in terms of contract type preference, i.e. agents who belong to these three archetypes will behave similarly, whereas hedonic agents will act differently. As such, more technologically advanced contracts that would yield higher energy and financial savings, would also result in higher CO_2 savings and increased social welfare.

2.6 Model boundaries

The SIMP model provides insight into diffusion patterns of energy services (represented by a contract) and associated switching rate among predefined set of contracts offered by a retailer. Each time the agent decides to opt for a new contract, she gets the same types of contracts, i.e. there are no new types of contracts available on the market. In reality, retailers may develop new contracts on the basis of the current market share, thereby reflecting consumers' preferences and attitudes towards the contracts currently available, which may also result in some contract types disappearing over time, while others persisting for longer period. In this case, we could consider multiple retailers with different marketing strategies, based on specific consumers' characteristics.

The experience the agents get with each contract is modelled as exogenous variable, each time step randomly drawn from a predefined set of values for each performance indicator, whereas the evaluation of the current contract shall reflect upon learning effects from past experiences and adapt the current experience accordingly (e.g. through adaptive set for each indicator).

Additionally, the attitude threshold and techno-tolerance threshold are also exogenous, fixed at the initialization of the model. Fixed techno-tolerance threshold means that consumers disregard the contracts that are below their techno-tolerance threshold every time they opt for a contract change. Such an approach prevents the agent to consider more "technologically risky" contracts at the expense of better outcome (in terms of energy savings, environmental impact, etc.). Some consumers might be more "tech-savvy" and thus be less concerned with technological risk than others. Also, the perception for more "technologically risky" contracts may change over time, owing to the experience an agent has with a specific contract, which can ultimately result in adaptive techno-tolerance threshold. Similarly, attitude threshold shall consider agents' learning and adaptation and therefore be reflexive and reactive to the environment.

Though the model is named Subjective Individual Model of Prosumers, we currently model consumers' behaviour and do not consider self-generation. Future developments will extend the current model to include engagement strategies and energy services related to self-generation and self-consumption.

3 An integrated approach to consumer, retail market and electricity network

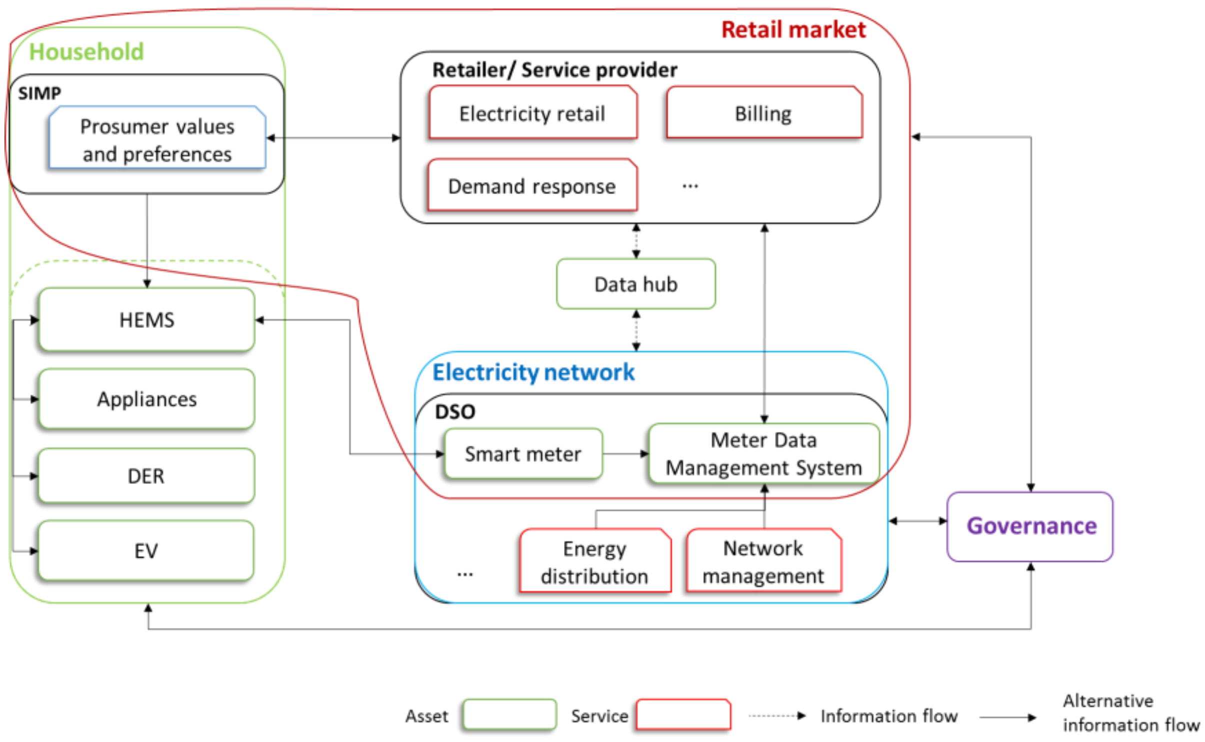
The Subjective Individual Model of Prosumer models the interaction of the electricity consumer with smart metering technologies and thereby enabled services, represented by electricity contracts. The electricity consumer is modelled as an individual and social agent who, through her interaction with the electricity retailer and the DSO, can affect the development of the future retail market.

The analysis of how the consumer/prosumer interacts with the electricity network and the market is of relevance for the current development of the EU internal energy market.

For this reason, our aim is to integrate SIMP with the physical grid and the retail energy market. We foresee a modular architecture (Figure 5), where we will be able to simulate the interaction of several actors (agents) communicating among each other and impacting each other's actions. This integrated model includes four modules and each module can involve several actors:

- **The household** (including different consumers/prosumers);
- **The retail market** (including consumers, prosumers, retailers, different types of service providers, electricity producers, DSO as neutral market facilitator, etc.);
- **The electricity network** (including the DSO and the TSO, as responsible for operation of the distribution and the transmission network, respectively);
- **The governance** (including regulatory authorities, local energy agencies, municipalities, etc.).

Figure 5. Integrated framework for consumer, retail market and electricity network



Source: own elaboration

The *household module* defines a set of rules associated with the load control mechanisms and linked with the consumers' behavioural patterns. This set of rules is partly defined in a contract, between the consumer and a retailer, or alternatively between the consumer and a service provider (e.g. aggregator, ESCO, etc.). These set of rules are linked with the consumers' values and preferences. In turn, these preferences affect both the contract choice (SIMP), but also the consumption profile, as they are encoded into the Home Energy Management System (HEMS). Also, HEMS integrates the set-points of different household appliances, distributed energy resources (photovoltaic panels, residential storage, etc.), electric vehicle, etc. and thus, it serves as interface for provision of different energy services (e.g. demand response). The output of this module is the evolution of consumption/production profiles under different consumers' behaviours.

The *retail market module* defines rules of the market dynamics among a multitude of actors, both traditional and new actors entering the scene. The rules for market functioning are defined by the regulator, which plays a key role in promoting well-functioning retail market. The output of this module is twofold: 1) market share of electricity retailers and service providers, and 2) evolution of contracts with innovative services beyond electricity supply (e.g. flexibility). In most of the EU Member States, electricity contracts are offered to the consumer by a retailer as a single contact point, that is also responsible for billing, whereas the DSO operates as an interface between the economic transactions and the physical network, playing the role of a neutral market facilitator. Different data management models are largely discussed in Europe (CEER, 2016) and in most of the EU countries, smart metering data are managed by the DSO, which then, upon consumer's consent, communicates these data to the retailer and any third party. Alternatively, there is an increasing trend in EU towards centralised data management model, by using a data hub (Figure 5).

In this context, different service providers (e.g. aggregators, ESCOs, etc.) may act on behalf of some consumers/prosumers and provide services to the retailer (e.g. portfolio balancing) or the DSO (congestion management, voltage support, etc.). Alternatively, retailers may enter in direct contact with the consumers/prosumers and provide different data analytics and comparison tools for increasing awareness of electricity use and request different services (e.g. demand response via variable electricity prices). As a result, following the consumers' preferences on contract choice (SIMP), retailers may come up with targeted marketing strategies and contracts for different types of consumers. At the same time, this will affect the market share of specific contracts or retailers. At the end, such evolution of contract's share and electricity consumption profiles will impact the operation of the electricity network and consequently necessitate a coordinated approach between the market and the electricity network operation.

The *electricity network module* represents the physical electricity grid with all the loads, power plants, lines, buses and other network equipment, where the magnitude of production and consumption must essentially match at all points in time. Technically, this is accomplished by the ramping up and down of variable output generation units, which in presence of growing RES penetration proves each time more challenging. Additionally, this may result in stretching the capacity of some power lines close to their limits or pose a significant challenge on the voltage, particularly in the low voltage area of the distribution network. As a result, the need for flexibility, both at the demand and the generation side, increases, which calls for development of innovative flexibility services, specified under a contract between the consumer/prosumer and the DSO/service provider. For that reason, this module also communicates to the SIMP (through the retailer or service provider) and as such, it can have effect on and be affected by the consumer preferences for electricity service contracts. In most of the EU Member States, the DSO owns and operates the smart meter (SWD, 2014). The metered data are then collected, validated and processed by the Meter Data Management System and used for different applications, such as billing, demand response, outage management, workforce management, etc. Alternatively, the metered data can be sent directly to an independent data hub.

This module returns network load as an output, and specifically, the need for different network management services, such as voltage support, congestion management, etc.

The *governance module* represents an overarching module, where public policies are shaped and implemented and consequently, it has impacts and is impacted by the other modules.

References

CEER. (2016). *Review of current and future data management models.*

Gangale, F., Mengolini, A., & Onyeji, I. (2013). Consumer engagement: An insight from smart grid projects in Europe. *Energy Policy, 60*, 621-628.

Gilbert, N. (2008). *Agent-based models.* Sage Publications.

Mengolini, A. (2017). *Prosumer behaviour in emerging electricity systems.* Politechnico di Torino.

SWD, E. C. (2014). *Benchmarking smart metering deployment in the EU-27.* Brussels: European Commission.

van Dam, K. H., Nikolic, I., & Lukszo, Z. (2017). *Agent-based modelling of socio-technical systems .* Springer.

Vasiljevska, J., Douw, J., Mengolini, A., & Nikolic, I. (2017). An agent-based model of electricity consumer: smart metering policy implications in Europe. *Journal of Artificial Societies and Social Simulation, 20*(1).

Wilensky, U. (1999). *Netlogo*. Retrieved from https://ccl.northwestern.edu/netlogo/

List of abbreviations and definitions

DER	Distributed Energy Resources
DSO	Distribution System Operator
ESCO	Energy Service Companies
EV	Electric Vehicle
GUI	Graphical User Interface
HEMS	Home Energy Management System

List of figures

List of tables

Annex: SIMP source code

Creating the model ("setup" module):

to **setup**
clear-all
set evaluation-period 120
set EGOISTIC 0
set HEDONIC 1
set BIOSPHERIC 2
set ALTRUISTIC 3
init-contracts
init-prosumers
check-archetype-shares
reset-ticks
clear-all-plots
plot-outcomes
end *(end of* ***'setup'*** *module)*

"setup" procedures

to **init-contracts**
(*set the initial values of the contracts for each performance indicator, i.e. communicated-outcome – ego: euros saved per month; hedo: percentage increase of comfort; bio: reduced kg* CO_2 *per month; alt: percentage increase of social welfare*)
if policy = 0 [
set communicated-outcomes matrix: from-row-list
[0.54 0.66 0 0.1 1 2 0.1 0.2] [*contract B*]
[0.54 0.66 0 0.1 1 2 0.1 0.2] [*contract C*]
[2.2 2.6 -12 -8 5 7 0.2 0.3] [*contract E*]
[3.2 4.0 -12 -8 8 10 0.3 0.4] [*contract F*]
[5 7.0 -17 -13 12 16 0.4 0.5]] [*contract G*]
(*set the technological risk for each contract*)
set techno-risks (list 1 2 4 5 6)]
if policy = 1 or policy = 2
[set communicated-outcomes matrix: from-row-list
[0 0.1 0 0.1 0 0.5 0.1 0.2] [*contract A*]
[0.54 0.66 0 0.1 1 2 0.1 0.2] [*contract B*]
[0.54 0.66 0 0.1 1 2 0.1 0.2] [*contract C*]
[1.08 1.32 -6 -4 2 4 0.2 0.3] [*contract D*]
[2.2 2.6 -12 -8 5 7 0.2 0.3] [*contract E*]
[3.2 4.0 -12 -8 8 10 0.3 0.4] [*contract F*]
[5 7.0 -17 -13 12 16 0.4 0.5]] [*contract G*]
(*set the technological risk for each contract*)
set techno-risks (list 1 1 2 3 4 5 6)]
(*set global variable with the dimensions of the communicated outcomes*)
let dimensions matrix: dimensions communicated-outcomes
set nr-of-contracts item 0 dimensions
set nr-of-perf-ind (item 1 dimensions) / 2
(*initialize the communicated-outcomes for later use*)
set extremes-of-communicated-outcomes matrix:from-row-list (n-values 1 [n-values (nr-of-perf-ind * 2) [-100]])

ask patches with [pxcor = 0] [set pcolor grey]
ask patches with [pxcor = 1] []
ask patches with [pxcor = 2] [set pcolor grey]
ask patches with [pxcor = 3] []
ask patches with [pxcor = 4] [set pcolor grey]
ask patches with [pxcor = 5] []
ask patches with [pxcor = 6] [set pcolor grey]
(*get the highest and lowest value of all performance indicators, i.e. communicated-outcomes for all contracts*)
let pi-nr 0
while [pi-nr < nr-of-perf-ind]
(*get the lowest lower boundary, and the highest higher boundary across all contracts for the performance indicator with pi-nr*)
(*start by getting the values of contract 0*)
[let contract-nr 0
let min-outcome matrix: get communicated-outcomes contract-nr (pi-nr * 2)
let max-outcome matrix: get communicated-outcomes contract-nr (pi-nr * 2 + 1)
(*iterate over the rest of the contracts to see whether they have higher maximums or lower minimums*)
set contract-nr contract-nr + 1
while [contract-nr < nr-of-contracts]
[set range matrix: get communicated-outcomes contract-nr pi-nr
if matrix: get communicated-outcomes contract-nr (pi-nr * 2) < min-outcome [
set min-outcome matrix: get communicated-outcomes contract-nr (pi-nr * 2)]
if matrix: get communicated-outcomes contract-nr (pi-nr * 2 + 1) > max-outcome [set max-outcome matrix: get communicated-outcomes contract-nr (pi-nr * 2 + 1)]
set contract-nr contract-nr + 1]
matrix: set extremes-of-communicated-outcomes 0 (pi-nr * 2) min-outcome
matrix: set extremes-of-communicated-outcomes 0 (pi-nr * 2 + 1) max-outcome
set pi-nr pi-nr + 1]
normalize-communicated-outcomes
end (*end of procedure '**init-contracts***)

to **init-prosumers**
(*prosumers initially having contract A*)
create-prosumers (prosumers-contract-A) [set current-contract-nr A]
(*prosumers initially having contract B*)
create-prosumers (prosumers-contract-B) [set current-contract-nr B]

Providing the main time step ("go" module):

to **go**
show-debug-message word "*** TICK *** " ticks -1
ask prosumers [
set changed-contract? false
(*agents get experience with the current contract*)
calc-experienced-outcomes
(*normalisation of the outcomes from the current contract*)
normalise-experienced-outcomes
(*agents built attitude towards the current contract*)

calc-attitudes
let satisfied? am-i-satisfied
let contract-ended? (elapsed-contract-time >= contract-duration)
(*if the agent is not satisfied with the current contract* OR if the contract has ended)
ifelse not satisfied? or contract-ended?
[let loop-contract-nr 0
    while [loop-contract-nr < nr-of-contracts]
    (*iterate over available contracts to ask neighbours' opinion*)
    [**ask-neighbours-opinions loop-contract-nr**
    (*disregard the contracts that have a techno-tolerance value higher than the agent's own one*)
    (*first determine which contract is the least 'technologically invasive'*)
    let min-techno-risk min techno-risks
    let least-risky-contract position min-techno-risk techno-risks
    if techno-tolerance-threshold < item loop-contract-nr techno-risks
    (*leave this contract out of the consideration if it is not the least 'technologically invasive' one*)
    [if loop-contract-nr != least-risky-contract
    [set influenced-expected-general-attitudes replace-item loop-contract-nr influenced-expected-general-attitudes -100
    set loop-contract-nr loop-contract-nr + 1]
*(*insert a value of *-100 in the influenced-expected-general-attitudes if the contract is not considered*)
if not satisfied?
(*leave the current contract out of the consideration*)
[set influenced-expected-general-attitudes replace-item current-contract-nr influenced-expected-general-attitudes -100]
ifelse max influenced-expected-general-attitudes != -100
(*choose the offer with the highest influenced-expected-general-attitudes*)
[**choose-contract**
set elapsed-contract-time 0]
[if contract-ended?
[set elapsed-contract-time 0]]]
(*if prosumer is satisfied and the contract did not end*)
[set elapsed-contract-time elapsed-contract-time + 1]
calculate-satisfactions
] (*end of procedure '**ask prosumers**'*)
plot-outcomes
tick
(*for Policy = 2: Change environmental weights of all agents, 4 years after the start of the simulation*)
if ticks = 40 and policy = 2
[**do-environmental-campaign**
]
if ticks >= evaluation-period
[stop]
end *(end of* ***'go'*** *module)*

"go" procedures
(*agents get experience with the current contract*)

to **calc-experienced-outcomes**
let pi-nr 0
(*iterate over the performance indicators, i.e. outcomes*)
while [pi-nr < nr-of-perf-ind]
(*calculate the experienced outcome*)
[let min-outcome matrix: get communicated-outcomes current-contract-nr (pi-nr * 2)
let max-outcome matrix: get communicated-outcomes current-contract-nr (pi-nr * 2 + 1)
let outcome min-outcome + random-float (max-outcome - min-outcome)
(p*ut the outcome in the list for the current time step*)
set experienced-outcomes replace-item pi-nr experienced-outcomes outcome
(*add the outcome to the total outcomes*)
set total-outcomes replace-item pi-nr total-outcomes ((item pi-nr total-outcomes) +outcome)
set pi-nr pi-nr + 1]
end (*end of procedure '**calc-experienced-outcomes**')*

(*normalisation of the outcomes from the current contract*)
to **normalise-experienced-outcomes**
let loop-pi-nr 0
let normalised-experienced-outcomes n-values nr-of-perf-ind [-100]
for each experienced-outcomes
[let experienced-outcome ?
let normalised-experienced-outcome normalise loop-pi-nr experienced-outcome
set normalised-experienced-outcomes replace-item loop-pi-nr normalised-experienced-outcomes normalised-experienced-outcome
set loop-pi-nr loop-pi-nr + 1]
(*put the normalised experience with the current contract into the memory*]
matrix:set-row normalised-experienced-outcomes-memory current-contract-nr normalised-experienced-outcomes
end (*end of procedure '**normalise-experienced-outcomes**')*

(*agents built attitude towards the current contract*)
to **calc-attitudes**
let normalized-experienced-outcomes matrix:get-row normalized-experienced-outcomes-memory current-contract-nr
(*calculate agent's attitude towards the current contract and store it*)
let attitudes-current-contract calc-contract-attitudes normalized-experienced-outcomes
(*put the attitudes into the attitudes matrix*)
matrix:set-row attitudes current-contract-nr attitudes-current-contract
(*calculate the general attitudes*)
let general-attitude sum (matrix:get-row attitudes current-contract-nr)
(*insert the general attitude of the current contract into the list of general attitudes*)
set general-attitudes replace-item current-contract-nr general-attitudes general-attitude
end (*end of procedure '**calc-attitudes**')*

to evaluate-contract [contract-nr]
(*get the normalized average communicated outcomes*)

let temp-communicated-outcomes matrix:get-row normalised-avg-communicated-outcomes contract-nr
(*calculate attitudes*)
let attitudes-temp-contract calc-contract-attitudes temp-communicated-outcomes
(*put these attitudes into the prosumers-own matrix of expected attitudes*)
matrix:set-row own-expected-attitudes contract-nr attitudes-temp-contract
end

(*iterate over available contracts to ask neighbours' opinion*)
to **ask-neighbours-opinions** [contract-nr]
(*sum the attitudes of different neighbours*)
let summed-neighbours-attitude 0
let nr-of-neighbours-with-experience 0
(*initialise provisional value for the number of contacts*)
let nr-of-contacts -100
let loop-nr-of-contacts 0
while [loop-nr-of-contacts < nr-of-interactions]
[ifelse random-float 1 > heterogeneity
[ask n-of 1 prosumers with [[archetype] of self = [archetype] of myself]
(*if the prosumer has experience*)
[if item contract-nr general-attitudes > -100
set nr-of-neighbours-with-experience nr-of-neighbours-with-experience+1
(*get the neighbours' attitude for this contract*)
let neighbours-attitude item contract-nr general-attitudes
(*add the neighbour's attitude to the summed attitudes*)
set summed-neighbours-attitude summed-neighbours-attitude + neighbours-attitude]
]]
(*introduce heterogeneity, i.e. ask opinion from agents belonging to archetype different than the agent's own*)
[ask n-of 1 prosumers with [[archetype] of self != [archetype] of myself]
[if item contract-nr general-attitudes > -100
(*if the prosumer has experience*)
set nr-of-neighbours-with-experience nr-of-neighbours-with-experience + 1
(*get the neighbours' attitude for this contract*)
let neighbours-attitude item contract-nr general-attitudes
(*add the neighbour's attitude to the summed attitudes*)
set summed-neighbours-attitude summed-neighbours-attitude + neighbours-attitude]
]]
set loop-nr-of-contacts loop-nr-of-contacts + 1]   (end of 'while' loop)

(*initialise provisional value*)
let influenced-expected-general-attitude -100
(*if there is at least 1 neighbour with experience*)
ifelse nr-of-neighbours-with-experience > 0
(*calculate the average neighbours' general attitude for the respective contract*)
[let avg-neighbours-attitude summed-neighbours-attitude / nr-of-neighbours-with-experience
(*create a temporary variable with the own expected attitude for the respective contract*)
let own-expected-general-attitude item contract-nr own-expected-general-attitudes

set influenced-expected-general-attitude own-expected-general-attitude + (avg-neighbours-attitude - own-expected-general-attitude) * susceptibility]
(*if there are no neighbours with experience take agent's own expected attitude*)
[set influenced-expected-general-attitude item contract-nr own-expected-general-attitudes]
set influenced-expected-general-attitudes replace-item contract-nr influenced-expected-general-attitudes influenced-expected-general-attitude
end (*end of procedure '***ask-neighbours-opinions***')*

(*agents choose new contract*)
to **choose-contract**
let old-contract-nr current-contract-nr
(choose the contract with the best general attitude from the available contracts)
let max-general-attitude max influenced-expected-general-attitudes
set current-contract-nr position max-general-attitude influenced-expected-general-attitudes
(*update the visual representation considering the new chosen contract*)
set xcor current-contract-nr
(acknowledge if contract change took place)
if old-contract-nr != current-contract-nr [
set changed-contract? true]
end (*end of procedure '***choose-contract***')*

(*how satisfied the agent is with the choice of the new contract – i.e. how close the new contract matches consumer preferences*)
to **calculate-satisfactions**
set attitude-satisfaction (item current-contract-nr general-attitudes) - threshold-attitude
set techno-tolerance-satisfaction techno-tolerance-threshold - (item current-contract-nr techno-risks)
end (*end of procedure '***calculate-satisfactions***'*)

(*introduce an environmental campaign*)
to **do-environmental-campaign**
let percentage-increase 2
(*100% increase of agents biospehric weights*)
ask prosumers [
let bio-weight item BIOSPHERIC weights
set bio-weight bio-weight * percentage-increase
set weights replace-item BIOSPHERIC weights bio-weight
(*re-normalize the weights by dividing each of them by the total*)
set weights (map [? / sum weights] weights)
(*the archetype might have changed, see what is the current maximum*)
let max-weight max weights
set archetype position max-weight weights
if archetype = EGOISTIC [set color red]
if archetype = HEDONIC [set color blue]
if archetype = BIOSPHERIC [set color green]
if archetype = ALTRUISTIC [set color pink]]
end (*end of procedure '***do-environmental-campaign***'*)

(*prosumers initially having contract C*)
create-prosumers (prosumers-contract-C) [set current-contract-nr C]
(*prosumers initially having contract D*)
create-prosumers (prosumers-contract-D) [set current-contract-nr D]
(*prosumers initially having contract E*)
create-prosumers (prosumers-contract-E) [set current-contract-nr E]
(*prosumers initially having contract F*)
create-prosumers (prosumers-contract-F) [set current-contract-nr F]
(*prosumers initially having contract G*)
create-prosumers (prosumers-contract-G) [set current-contract-nr G]
(*introduce randomness in the archetype determination*)
let alternative-prosumer-nr 0
ask prosumers
(*initialise total-outcomes to zero*)
[set total-outcomes n-values nr-of-perf-ind [0]
(*visualise the prosumers*)
setxy current-contract-nr random-ycor
set shape "house"
(*There is not yet experience to compare with the threshold, so we presume satisfaction*]
set satisfied 1
(*determine the archetype of each prosumer based on the slider values*)
let archetype-determinant alternative-prosumer-nr / count prosumers
ifelse archetype-determinant < share-ego [
set techno-tolerance-threshold (random-float 10) + 1
set archetype EGOISTIC
set color red]
[ifelse archetype-determinant >= share-ego and archetype-determinant < share-ego + share-hedo [
set techno-tolerance-threshold (random-float 10) + 1
set archetype HEDONIC
set color blue]
[ifelse archetype-determinant >= share-ego + share-hedo and archetype-determinant < share-ego + share-hedo + share-bio [
set techno-tolerance-threshold (random-float 10) + 1
set archetype BIOSPHERIC
set color green]
[if archetype-determinant >= share-ego + share-hedo + share-bio [
set techno-tolerance-threshold (random-float 10) + 1
set archetype ALTRUISTIC
set color pink]]
]]
(*initialise weights for each of the 4 archetypes (ego, hedo, bio, alt) to a random number between 0 and 1*)
set weights n-values nr-of-perf-ind [random-float 1]
(*normalize the weights by dividing each of them by the sum*)
set weights (map [? / sum weights] weights)
(*swap the maximum weight with the weight of the archetype*)
let max-weight max weights
let max-index position max-weight weights
let archetype-weight item archetype weights

set weights replace-item max-index weights archetype-weight
set weights replace-item archetype weights max-weight
(*initialize outcomes, attitudes, etc.*)
set normalized-experienced-outcomes-memory matrix: from-row-list (n-values nr-of-contracts [n-values nr-of-perf-ind [-100]])
set attitudes matrix: from-row-list (n-values nr-of-contracts [n-values nr-of-perf-ind [-100]])
set own-expected-attitudes matrix: from-row-list (n-values nr-of-contracts [n-values nr-of-perf-ind [-100]])
set general-attitudes n-values nr-of-contracts [-100]
set own-expected-general-attitudes n-values nr-of-contracts [-100]
set influenced-expected-general-attitudes n-values nr-of-contracts [-100]
set experienced-outcomes n-values nr-of-perf-ind [-100]
(*set susceptibility equal to a value drawn from a uniform distribution between min and max – min and max are sliders*)
set susceptibility min-susceptibility + random-float (max-susceptibility - min-susceptibility)
(*calculate the own expected general attitudes from the communicated outcomes*)
let loop-contract-nr 0
    while [loop-contract-nr < nr-of-contracts]
    [evaluate-contract loop-contract-nr
    (*calculate the general attitudes for each contract*)
    let summed-attitudes sum (matrix: get-row own-expected-attitudes loop-contract-nr)
    (*general-attitude is the average of the four attitudes*)
    let general-attitude (summed-attitudes / nr-of-perf-ind)
    set own-expected-general-attitudes replace-item loop-contract-nr own-expected-general-attitudes general-attitude
    set loop-contract-nr loop-contract-nr + 1]
    (*no contract is changed at the initial step*)
    set changed-contract? False
    set alternative-prosumer-nr alternative-prosumer-nr + 1]
end (*end of procedure '**init-prosumers***)

to **check-archetype-shares**
let total-shares share-ego + share-bio + share-hedo + share-alt
        if total-shares != 1 [
        error "The total shares of the archetypes do not equal 1"
        stop]
end (*end of procedure '**check-archetype-shares***)

KJ-NA-29514-EN-N

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

Joint Research Centre

EU Science Hub

Publications Office

doi:10.2760/786611

ISBN 978-92-79-98285-9