AGADE Using Personal Preferences and World Knowledge to Model Agent Behaviour

Thomas Farrenkopf[1(✉)], Michael Guckert[1], and Neil Urquhart[2]

[1] KITE - Kompetenzzentrum für Informationstechnologie, Technische Hochschule Mittelhessen, Giessen, Germany
{thomas.farrenkopf,michael.guckert}@mnd.thm.de
[2] School of Computing, Edinburgh Napier University, Edinburgh, Scotland
n.urquhart@napier.ac.uk

Abstract. BDI agents provide a common well established approach for building multi-agent simulations. In this paper we demonstrate how semantic technologies can be used to model agent behaviour. Beliefs, desires and intentions are mapped flexibly to corresponding OWL ontologies structured in layers. This reduces JAVA coding efforts significantly. Reasoning mechanisms and rule evaluation are used to compute agent behaviour by deriving an agent's actions from declaratively formulated rules. An agent's knowledge of its environment and its personal preferences can be expressed and human behaviour can be simulated. The approach is implemented in an integrated tool for running round based agent simulations (AGADE).

Keywords: Multi-agent system · BDI · OWL ontology · Market simulation · Human behaviour

1 Introduction

Multi-agent simulations are a powerful tool for the analysis of complex adaptive systems consisting of independent individuals [9]. These individuals are modelled as agents and their individual behaviour leads to emergent patterns of behaviour in the community of agents. Typical applications are market simulations and predictive investigations of organisational development.

AGADE (Agile Agent Development Environment) a tool for round-based multi-agent simulations where each agent is equipped with world knowledge coded in a layered ontology was developed at Technische Hochschule Mittelhessen. Moreover AGADE allows the specification of a social structure for the community of agents i.e. a sociogram. Agents can e.g. inhabit a community that follows rules of small world networks. Information about this structure is made available to the agents so that they may be aware of their position and their importance in their social environment. A modified version of the page rank algorithm [14] is used to calculate an influence matrix that quantifies mutual influence [7, pp.240–241]. Other social structures can be generated by the tool as well. AGADE is highly configurable and can be used to run different scenarios [7].

Y. Demazeau et al. (Eds.): PAAMS 2015, LNAI 9086, pp. 93–106, 2015.
DOI: 10.1007/978-3-319-18944-4_8

In this paper we address the principal approach of using OWL (Web Ontology Language) ontologies to model agent behaviour i.e. how to code an agent's knowledge and its preferences using OWL and SWRL (Semantic Web Rule Language) and how this corresponds to the BDI model. A major benefit of this approach is the reduction of programming efforts and a clearer separation of concerns in the overall simulation model. The approach is implemented in the tool AGADE mentioned above. Details will be discussed in this paper and examples and demonstrations will be presented.

2 Motivation

This research aims at building stronger connections between semantic technologies and multi-agent systems that can be used and reused flexibly for different scenarios. We demonstrate the use of semantic technologies for modelling realistic purchasing decisions of buyers in simulated market places. Such simulations may be used to enhance business games and potentially within business decision support systems. Thorough literature studies have shown that the idea of using ontologies is not entirely new but up to now these approaches have not lead to a really integrated solution. We refer to [7] for this discussion.

AGADE has been used to run simulations on a model of a mobile phone market where buyers often base their buying decision on social influence. Therefore the agents were modelled to follow the pattern of opinion leadership and the market development indeed developed as predicted and produced the expected statistics [7]. This was a proof of concept and now we aim at modelling more complex scenarios with a more heterogeneous structure of market participants. This presents the challenge of having to model different behavioural patterns into our agents. Besides varying problem solving patterns (How does the agent perform a buying decision?) we also have to model differing personal preferences (What are the agent's personal preferences concerning mobile phones?). This work can be simplified if we separate the agent's Java implementation from the definition of the behavioural patterns.

3 Agents and Ontologies

According to classical definitions an agent is an autonomous software entity which observes its environment, reacts to impulses (internal or external) and acts independently within a defined environment. External stimuli and available information are used to determine an agent's actions. Agents focus these actions on reaching given goals while following available plans. Newell and Simon [13] have already coined the term intelligent agent in 1972 for such an entity. A common paradigm for the development of intelligent agents is the so called BDI concept [5]. The acronym BDI represents three aspects that define the characteristics of an agent: beliefs, desires and intentions. A BDI agent has knowledge about its world (beliefs) and pursues goals (desires) while following given strategies (intentions). Therefore the agent belief base stores everything an agent knows (or believes to know) about the environment it lives and acts in. Here the things that exist and relations between these things can be specified i.e. domain knowledge is made available to the agent. In classical BDI implementations using frameworks like Jadex all aspects have to be coded in Java classes fitting into the hotspots of the framework [4]. While the basic flow of control is left in Java classes we shift certain aspects of the agent so that we can use declarative rule languages. This is described in detail in section 4.

Today we have widely standardised formalisms to represent knowledge in what we call ontologies and we will use the standardised techniques to model BDI agents. Formally an ontology $\mathcal{O}$ is a triple $(\mathcal{C}, \mathcal{R}, \mathcal{I})$ where $\mathcal{C}$ is a set of concepts, $\mathcal{R}$ a set of relations, and $\mathcal{I}$ a set of individuals. Concepts formally denote sets of individuals: sets of individuals are the extension of concepts while concepts are the intentional representation of the corresponding sets of individuals. An individual that belongs to a concept is called an instance of that concept. The elements of $\mathcal{R}$ are relations (also called roles or object properties) having subsets of $\mathcal{C}$ as domain and range. The extension of a role is then a set of pairs (c, d) with $c, d \in \mathcal{I}$. Additionally individuals can have data properties where they get linked to primitive data e.g. strings or numbers. Typically ontologies are formulated by means of description logics with differing levels of expressiveness [2]. Usually description logics are proper subsets of first order logic where typically expressiveness has been traded for decidability. Inference knowledge is implicitly given by the underlying reasoning mechanisms of the available reasoning instruments. Here we use OWL and SWRL both specified by W3C [11],[15]. The sets of beliefs (i.e. knowledge), desires (i.e. goals) and intentions (i.e. plans

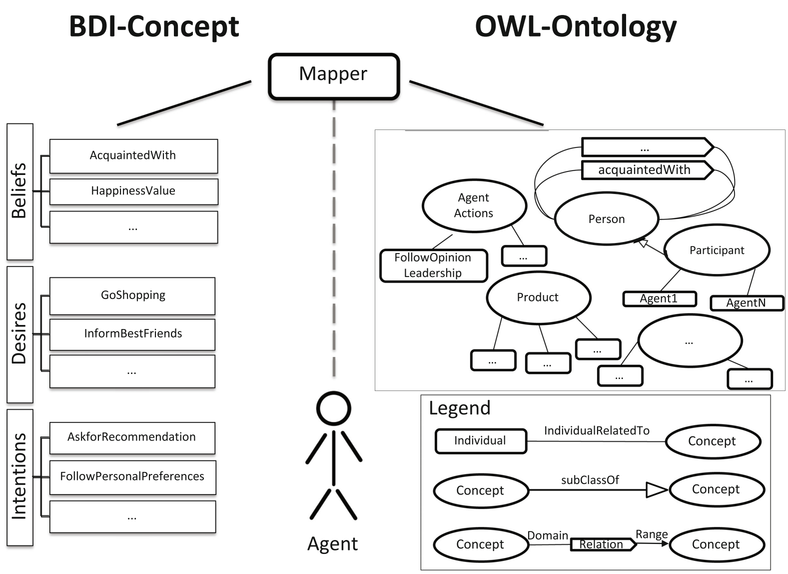

Fig. 1. OWL-BDI-Mapping

of how to reach the goals) of an agent are mapped into a layered OWL ontology. Desires and intentions correspond to OWL individuals of appropriate concepts and beliefs are represented by instantiations of relations (see Fig. 1). A *belief change listener* ensures that beliefs of an agent which are modified in Java operations that are part of the implementation will always be kept up-to-date in the ontology.

The ontology and its inference mechanisms are used to determine the behaviour of an agent e.g. rules are used to determine plans and calculate actions. Each agent has its own private ontology while we make sure that agents have a common understanding of the environment by providing commonly shared elements. We implement this using a layered approach we will discuss in the next section.

4 Layered Ontology

The development of a universally applicable integration of semantic technologies and agent based systems is still a challenge. Our idea is to achieve a blueprint for an architecture that can easily be adapted to various simulation scenarios. We propose a layered ontology (Fig. 2) where domain knowledge can be separated by its degree of generality. We distinguish between the abstract domain layer (ADL), the specific domain layer (SDL) and the individual domain layer (IDL). While ADL contains the most general knowledge elements, SDL can be used for more specific aspects. Individual knowledge is coded in IDL. ADL and SDL are shared by all agents leading to a common understanding of concepts, which realises an ontological commitment that enables communication among the agents. This approach leads to flexibility and a higher degree of reusability as at least ADL can be applied to a wide range of simulations of consumer product market places.

Considering the simulation of markets the abstract domain layer can describe general concepts, relations and individuals which are not restricted to a specific product market. SDL refines ADL by specialising abstract elements of ADL to fit the requirements of a specific market domain. The individuality of each agent

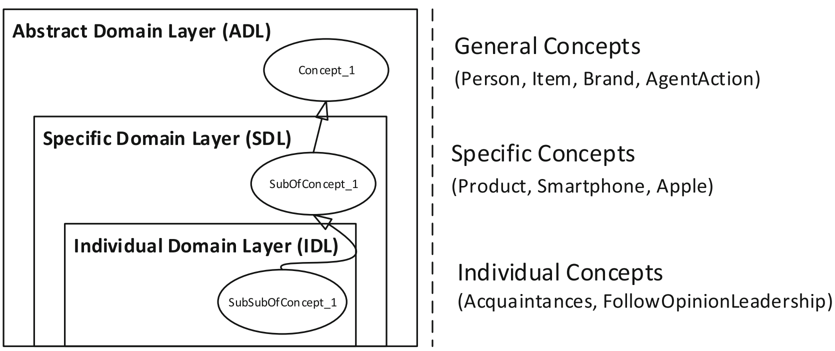

Fig. 2. Three-layer architecture

is expressed in IDL. It contains individual beliefs and definitions of individual behaviour of each agent (e.g. how an agent reacts to a certain stimulus). OWL allows ontologies to import other ontologies. We use this to import the general ontologies into the more specific ones. From a mathematical point of view the set of general concepts is a subset of the specific knowledge available to an agent.

The separation of knowledge into layers allows the general control of the simulation to be independent of specific terms of a given scenario. For example: creating a market simulation of a mobile phone market specific concepts and relations are modelled in SDL e.g. concepts like mobile phone, smartphone or touchphone. The individual aspects of an agent and how it in fact behaves in this market is expressed in the IDL e.g. that it follows opinion leadership.

This layered approach is mirrored into the Java application that implements the BDI concept. We developed an `AbstractOWLAgent` class that describes fundamental elements of an OWL-BDI agent that enable it to participate in AGADE simulations. References and methods to maintain ontologies and trigger plans are implemented here. We equip each agent with its own reasoner and private ontology which is accessed using the OWL API [10]. Subclasses of `AbstractOWLAgent` are on the level of SDL and specify more concrete aspects of an agent (see Fig. 3). Each subclass references an IDL which in turn is the key to the individual behaviour of an agent and describes the type of an agent as well. For example: AGADE has one general market participant class and distinguishes between seller or customer in the individual ontologies used in the prototypical mobile phone market implementation. All available agent actions of the specific market scenario (e.g. plans) have to be expressed as a member of the concept `AgentAction`. Specific plans are relevant for a specific scenario and therefore they are attributable to the SDL.

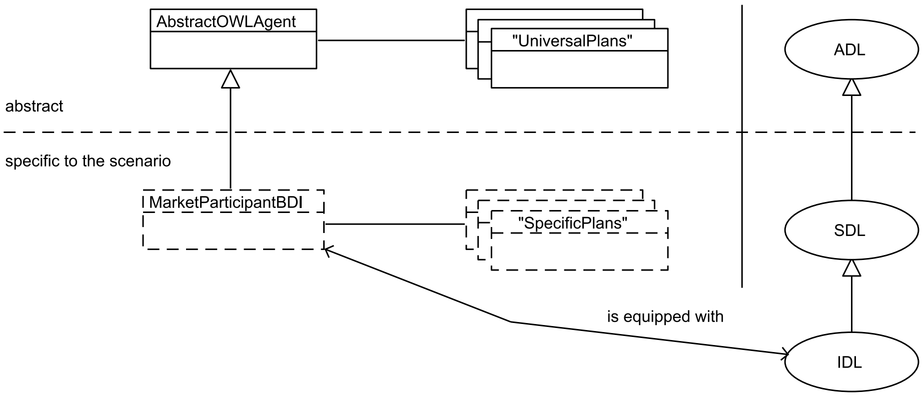

Fig. 3. Layered ontology and agent classes comparison

We decided to use a well established multi-agent framework as underlying technology because it does provide a set of convenience tools (logging, monitoring,...) and fully functioning BDI infrastructure. Jadex was the tool of choice because it is an easily available Java based solution that we achieved to seamlessly connect to the reasoning mechanisms of the ontologies and its reasoners.

In compliance with the Jadex framework possible individual actions have to be implemented as plans. This is done in the Java code that implements the agent. Plans in Jadex can be represented as methods inside the Java class that implements the agent or alternatively as plain Java classes which have to provide a so called plan body method. We recommend to code plans as Java classes to keep the agent behaviour pattern as flexible as possible, because plans written in Java classes can easily be made available to different agents by simply adding `@Plan` annotations to the specific agent class. AGADE can create *plan pools* out of available classes annotated as plans thus making them available in other simulations.

The Java classes annotated as plans find a corresponding member of concept `AgentAction` in the ontology. These links makes facts and rules of the ontology accessible to the agents. The object property `nextAgentAction` (with domain `Person` which is equivalent to the set of agents and range `AgentAction`) together with a rule determines how the agent decides which plan to chose next. The next agent actions are periodically triggered by the round based management of AGADE. Note that the ontology based belief base leads to a very flexible architecture, because important aspects of the agent do not have to be coded statically any more but may be expressed in the rules of the ontology.

Agent knowledge is limited to what is defined in the hierarchy of ontologies possibly differing from what other agents know. An agent may extend its knowledge base during a simulation meaning that it has learning capability. Agents communicate with other agents (e.g. they exchange information about product details) and this communication may refer to knowledge items that belong to the IDL layer. Therefore agents can exchange information which contains concepts that may be totally new for the receiving agent. The agent may then add new facts acquired through this information exchange into its belief base. When incorporating a new concept into its IDL the agent has to obtain all available information relating to that concept. SDL and ADL layers are shared among the agents so that a concept with a direct superclass in SDL or ADL can easily be added to the IDL of the learning agent. Otherwise, if the concept does not have direct ancestors in ADL or SDL the super classes of the sending agent must also be included. Individuals and facts (properties) about individuals can be added directly, if they are instances of a concept defined in ADL and SDL. But agents can also exchange definitions of concepts and information about individuals that are instances of concepts of an IDL. We currently expect that every concept in IDL is a subconcept of concepts in ADL – possibly transitively. This is ensured by a Java routine that performs validation checks on the ontologies. To summarise: Let o_1 and o_2 be individual ontologies. The intersection $o_1 \cap o_2$ is uncritical because it is obviously available to both agents. From the perspective of o_1 the set $o_2 \setminus o_1$ is critical, because it contains elements of $\mathcal{C}$, $\mathcal{R}$ or $\mathcal{I}$ which are relevant for the learning process.

For example: Each product p is represented as an instance of concept *Product*. Let the IDL of an agent a_1 contain product p and further assume that the IDL of agent a_2 does not contain p. If a_1 wants to communicate details of p to a_2 and p is totally new to a_2, the agent a_2 has to add p into its IDL. In this case the corresponding concept hierarchy will be added to the ontology of agent a_2 if necessary.

This learning capability has direct effects on the actions of agents e.g. their buying behaviour. The layered approach enables the learning capability possible described above.

5 Personal Preferences

In general market segments consist of buyers and sellers who demand and offer competing products. A customer will compare these products and try to rank them according to his personal preferences by considering characteristics of available products [3, pp.202-204]. This could possibly be retail prices or any technical features measured quantitatively e.g. camera resolution or battery life span of a mobile phone.

To enable multi criteria comparisons quantifiable attributes are normalised to percentages using the span between the highest and the lowest value that appears among the described products of one kind. For example: let the camera resolution values within a fictive mobile phone market segment range from a minimum value of 4.1 megapixels to a maximum of 20.7 (see Fig. 4). The normalised percentage value of a camera resolution of 15.9 megapixels is then calculated as follows: the actual difference between 15.9 and 4.1 ($15.9 - 4.1 = 11.8$) is divided by the difference between the maximum value of 20.7 and the minimum value of 4.1 ($20.7 - 4.1 = 16.6$): $\frac{11.8}{16.6} = 0.7108 \cdot 100 = 71,08\%$. With this percentage rate a camera resolution with 15.9 megapixels can be estimated to lie in the upper third quantile. But obviously consumers will base their buying decision not only on one attribute. Each consumer weighs different characteristics of a product with different importance. To take these individual preferences into account the criteria get weighted with weighting factors between 0 and 1 which sum up to 1. In our example we may weigh camera resolution with a factor of 0.3 leaving 0.7 for other attributes. The calculated percentage of 71.08 gets multiplied by the individual comparison factor of 0.3: $71.08 \cdot 0.3 \approx 21.33$. To summarise what we have just discussed: Let $a_1, ..., a_n$ be attributes of an object and p_i the corresponding calculated percentage values. The *weighted preference value* of that object is the sum $\sum_{i=1}^{N} w_i \cdot p_i$ where $\sum_{i=1}^{N} w_i = 1$. By definition it lies between 0 and 100.

These are personal preferences, therefore we implement them in the IDL of an agent. Personal preferences are an integral part of the decision process where one product is selected out of many. Another aspect of a buying plan we have to model is the acquisition of information that is input to the calculations of personal preferences.

A simple buying plan of an agent that follows personal preferences may consist of the following actions:

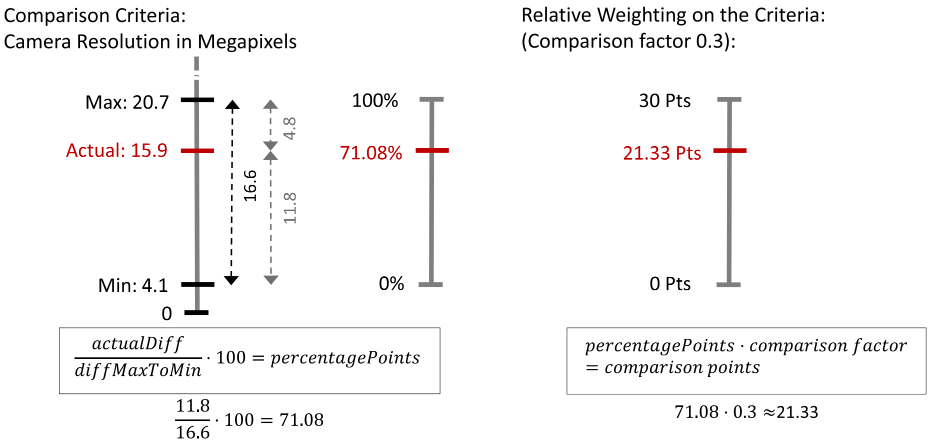

Fig. 4. Calculation of percentage points

1. select *persons* who own a phone
2. ask selected *persons* for all technical data of their phone
3. follow the personal preferences plan and determine the phone with the highest weighted preference value out of the set of phones collected in the step before and select it

In the following we show how such a buying plan can be expressed in the private ontology of an agent. Agents are represented as members of concept `Person` and everything that may be owned in some way or other by a person belongs to the concept `Item` . The properties `hasProduct`, `knows` and `hasAquaintanceValue` are elements of $\mathcal{R}$ with domain and range `Item` and `Person` respectively.

If an agent a_1 has a `knows` relation to another agent a_2 and a_2 `hasProduct` q and $q \in$ `Item` and q has attributes of a `Phone`, a_1 can conclude that a_2 is a person who owns something that is a phone. a_2 is classified as a member of concept `PersonWithItem` $\subset$ `Person` by using ontology reasoning techniques. Note that q does not have to be defined directly as a phone as the OWL reasoner will conclude this from the properties of q. If the IDL of an agent a does not contain a member of concept `Phone`, an information gathering process will be started. One way to get information about phones is picking agents from the direct social environment which belong to `PersonWithItem` and ask them for advice. As product comparison requires at least two items, information gathering is repeated until the agent knows at least two products. Alternatively agents can delegate a request to one of their neighbours i.e. all agents it is connected with or contact sellers (agents that are members of `Seller`) directly to get available products instead of asking other customers.

Data property relations are used for describing technical data of phones representing numerical values. SWRL math built-ins enable an OWL reasoner to perform mathematical operations and would be suitable for the calculation of personal preference values. However, the support of SWRL built-ins is limited.

Additionally built-ins may cause SWRL rules to become undecidable and therefore we implemented mathematical operations in Java and made them accessible for the ontology [8]. The mathematical operations are triggered by the rule evaluation process during the calculation of an agent's personal preferences. Relevant data will be retrieved from the agent's private ontology and gets updated immediately with the results calculated in Java (see Fig. 5).

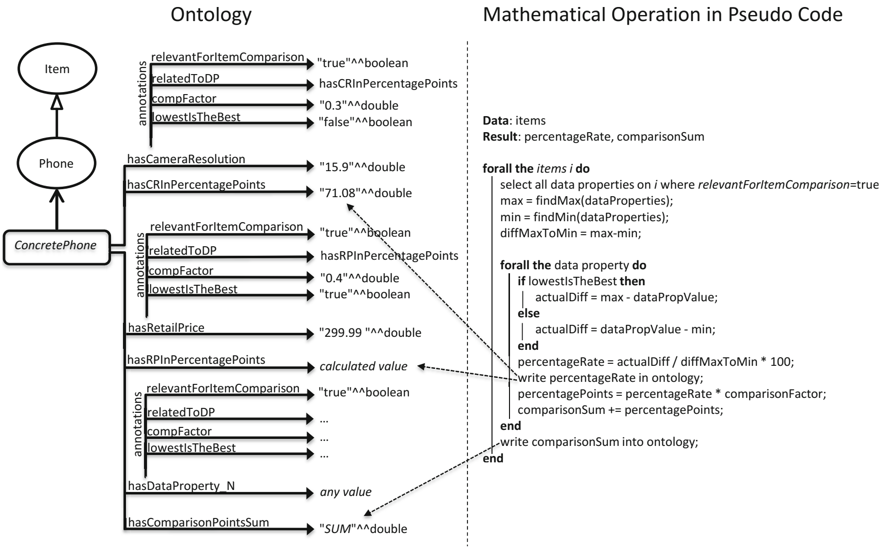

Fig. 5. Mathematical operations performed by a Java routine

The sum of each calculated comparison point is stored in a data property `hasComparisonPointsSum` related to the relevant item. The results are available to the reasoning process immediately. Mathematical operations are controlled by annotations in the ontology. Each element of the ontology can be annotated with instructions of how it will be handled during rule evaluation. In particular we designed *comparison annotations* which can be used to define how data properties will be evaluated during the calculation of personal preferences: `relevantForItemComparison`, `compFactor`, `relatedToDP` and `lowestIsTheBest`. They can be used for data properties of individuals of concept `Item`. While `relevantForItemComparison` carries a boolean value that indicates whether the property should be included in the calculation, `compFactor` contains the weighting factor. The `relatedToDP` annotation names another data property which stores the calculated percentage points. The `lowestIsTheBest` annotation changes the orientation of the comparisons: a lower value is considered better than a higher value. This applies to attributes as retail price or weight.

After the comparison process is finished, the agent is able to decide which item to buy. Additionally minimal requirements for a phone can be defined e.g.

the phone must have a battery life span that is as least as good as a given value. Such minimal requirements can be easily expressed with SWRL in the IDL layer as they do not require complicated calculations. Only those products which satisfy all given minimal requirements, are classified as members of the concept `ItemAccordingPreferences` and are then ranked according to personal preferences. If there is no item that matches the minimal requirements, the agent can search for further products by starting information gathering or alternatively reduce the minimal requirements. An example of a minimal requirement expressed with SWRL:

$$Phone(?x), double[>= 8.0](?y), hasCameraResolutionInMegapixels(?x, ?y) \\ \rightarrow ItemAccordingPreferences(?x)$$

Modelling personal preferences and including them in buying plans show how individual market behaviour can be expressed in an OWL ontology. The ontology is the main basis for the decision-making process of agents. Integration of Jadex agents and elements of the ontology is reached by use of annotations.

6 Results

After successfully simulating a homogeneous crowd of buyers acting in a mobile phone market [7, pp.245–247] where all agents follow a word of mouth decision process and depend on the advice of opinion leaders we modelled a more complex scenario where agents decide according to their personal preferences. Following different information acquisition plans e.g. *follow an expert* or *read test reports* the agents gather detailed information and use that as input to match the phones to their preferences. The agents inhabit a community that models social connections to fellow agents and to agents that represent phone sellers.

We collected data by running online surveys on a restricted group of persons (72 students and staff from Edinburgh Napier University). Among others the survey includes a quantitative analysis of the brand distribution, the brand loyalty of a person and the personal buying behaviour on which we will focus here. According to Holland we intentionally simplified our model by restricting it to a subset of available data [9, pp.45–46]. We aim at clarity and predictability by concentrating on a reduced set of facts.

Survey data is used to set comparison factors and minimal requirements for the products (see section 5). The question "Why did you choose this brand?" had seven possible answers, the following four were named the most often:

1. Decision based on test reports (34 persons)
2. Followed recommendation of friends and family (11 persons)
3. In-store consultation (3 persons)
4. Have many of my friends (socialisation) (2 persons)

Coming from a technically oriented organisation most of the participants used test reports as their main source of information. Test reports typically list all technical features of a product. We briefly describe the four behavioural patterns modelled. Details of the implementation were already discussed in section 5:

Table 1. Behavioural patterns

Buying behaviour	Description
Decision based on test reports	Every phone the agent is aware of is measured by personal preferences with respect to minimal requirements defined and the best is selected. If the agent does only know one phone or none it will gather information about phone models from every agent it is in social contact with and will apply personal preferences then.
Opinion Leadership	The agent will chose the phone that is possessed by the socially most important agent in its community (hub) (see [7])
In-store consultation	Some agents are modelled as phone sellers. An agent following this plan will contact a seller and apply personal preferences to each phone the seller recommends.
Have many of my friends	The agent will chose the phone that most of the agents he is socially connected to possess.

Mark that agents cannot take phone models from test reports as the simulation would converge very fast without influence of the environment. We consider social influence as very important and therefore the test reports are source of detail information only. Modelling these behavioural patterns and respective individual personal preferences resulted in 69 different IDLs. The survey in which participants were asked to order eight available attributes according to its subjective importance for them. The first four attributes were then weighted with the factors 0.4, 0.3, 0.2 and 0.1 meaning that only these four had an effect in the simulation. Non quantifiable factors cannot be used in the computation of the overall comparison factor and were therefore expressed as SWRL rules as the following that states that a phone should have an Android operating system:

$$
\begin{aligned}
&Android(?y), Smartphone(?x), hasOperatingSystem(?x, ?y) \\
&\qquad \rightarrow ItemAccordingPreferences(?x)
\end{aligned}
$$

If non quantifiable attributes were found among the first four attributes in an individual ordering the weighting factors were shifted so that the highest quantifiable factor received the value 0.4.

The phones modelled were taken from a web portal hosted by a popular German computer magazine [6]. Each brand in the simulation is represented by the product that was ranked highest by that portal (one for each brand). Smartphones are distributed uniformly over all agents at the beginning of the simulation.

We assume that communities generally follow a small world like structure [1]. Therefore we use Barabási's preferential attachment algorithm with slightly modified standard parameterisation to create the social environment the agents live in. The simulation was started with a uniform distribution of 1005 agents (15 for each of the four different buying behaviours listed above).

According to the stimulus-response model [12, p.24] agents need a trigger to start the buying process. We model this by using a so called happiness factor.

It is a numerical value the agent tries to maximise. This factor deteriorates continuously over time. Falling below a given threshold the factor indicates a lack of happiness and the agent gets active following its plans trying to make amends by looking for a new phone.

Fig. 6 shows the brand distribution after a simulation of 100 rounds. While the x-axis describes the number of rounds the y-axis shows the number of phones for a point in time.

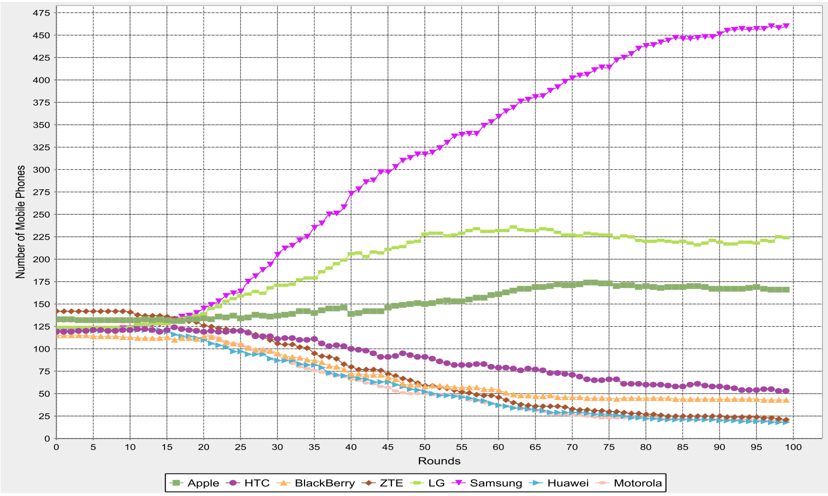

Fig. 6. Brand distribution chart after 100 rounds generated by AGADE

The simulation result and the survey data show significant similarities (Table 2 vs. 3). Looking at Apple, HTC, and LG differences can be observed. We see an explanation in the battery life span where there is a difference in the relevant models. 15 persons chose this attribute among the three most important criteria which caused a relatively high influence on the buying decision. As we simplified reality by only modelling one phone per brand we may have missed details that can cause this effect. Another aspect might be that an apparently rational buying decision based on facts and figures may mask rather subconscious elements of the decision that were not mentioned in the survey. Further research is necessary here.

All in all we have demonstrated that our approach can be calibrated to run realistic simulations on markets. Further research will be invested in even more elaborate behavioural patterns and alternative choices of attributes for the calculation of personal preferences to measure the sensitivity of the model.

Table 2. Brand distribution in survey

Brand	Distribution
Apple	21.74%
BlackBerry	4.35%
HTC	10.14%
Huawei	1.45%
LG	13.04%
Motorola	5.80%
Samsung	40.58%
ZTE	2.90%

Table 3. Brand distribution after 100 rounds

Brand	Distribution
Apple	16.52%
BlackBerry	4.28%
HTC	5.27%
Huawei	1.79%
LG	22.29%
Motorola	1.99%
Samsung	45.77%
ZTE	2.09%

7 Conclusion and Future Work

Using ontologies to model agents creates a new perspective for multi-agent simulation scenarios as programming details are reduced and the separation of modelling aspects from coding details is promising as scenarios can be set up with a reduced development effort. The ontology is used as a knowledge base and allows access to powerful standardised inference engines that offer leverage for the agents' decision processes. We define a three-layer ontology thus allowing agents to share knowledge and create a basic common understanding of their environment while enabling reuse of fundamental concepts. The generic approach will allow the simulation of different scenarios. We demonstrated buying behaviour as *personal preferences* can be modelled and how a heterogeneous community of buyers can be simulated with AGADE. The basic architecture with layered ontologies and its integration into the Java BDI application will be further elaborated and standardised. The degree of reuse that can be achieved will be investigated and formalised.

The simulation was run on a quad core CPU and 32GB RAM. We observed that the extensive use of ontologies results in a high memory consumption due to a large number of `String` objects used in the reasoning process and caching mechanisms of the OWL API. We can handle this issue by running simulations distributedly.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics **74**(1), 47–97 (2002)
2. Baader, F.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
3. Baker, M.: The Marketing Book. Taylor & Francis (2012)
4. Bellifemine, F., Caire, G., Greenwood, D.P.A.: Developing Multi-Agent Systems with JADE. Wiley, Chichester (2007)
5. Bratman, M.: Intention, plans, and practical reason. Center for the Study of Language and Information, Stanford and Calif (1999)

6. Chip: Vergleich: Handys im Test. http://www.chip.de/bestenlisten/Bestenliste-Handys-index/detail/id/900/ (Accessed, 10 January, 2015)
7. Farrenkopf, T., Guckert, M., Hoffmann, B., Urquhart, N.: AGADE. In: Müller, J.P., Weyrich, M., Bazzan, A.L.C. (eds.) MATES 2014. LNCS, vol. 8732, pp. 234–250. Springer, Heidelberg (2014)
8. Feng, Y., et al.: Discovering anomalies in semantic web rules. Secure System Integration and Reliability Improvement 33–42 (2010)
9. Holland, J.H.: Signals And Boundaries. MIT (2012)
10. Horridge, M., Bechhofer, S.: The owl api: A java api for owl ontologies. Semantic Web **2**(1), 11–21 (2011)
11. Horrocks, I., et al.: Semantic Web Rule Language (SWRL). W3C Member Submission (21 May 2004). http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/
12. Lantos, G.P.: Consumer Behavior in Action: Real-Life Applications for Marketing Managers. M. E. Sharpe Incorporated (2010)
13. Newell, A., Simon, H.A., et al.: Human problem solving, vol. 104. Prentice-Hall Englewood Cliffs, NJ (1972)
14. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
15. W3C OWL Working Group: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation, 11 December 2012. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/