# An agent architecture for knowledge discovery and evolution. ⋆

Tezira Wanyana[1,2][0000−0002−5139−8421] and Deshendran Moodley[1,2][0000−0002−4340−9178]

[1] Department of Computer Science, University of Cape Town (UCT), South Africa
[2] Center for Artificial Intelligence Research (CAIR), South Africa.
{twanyana, deshen}@cs.uct.ac.za

**Abstract.** The abductive theory of method (ATOM) was recently proposed to describe the process that scientists use for knowledge discovery. In this paper we propose an agent architecture for knowledge discovery and evolution (KDE) based on ATOM. The agent incorporates a combination of ontologies, rules and Bayesian networks for representing different aspects of its internal knowledge. The agent uses an external AI service to detect unexpected situations from incoming observations. It then uses rules to analyse the current situation and a Bayesian network for finding plausible explanations for unexpected situations. The architecture is evaluated and analysed on a use case application for monitoring daily household electricity consumption patterns.

**Keywords:** Agent architecture · BDI · Knowledge discovery and evolution · Abductive theory of method.

## 1 Introduction

With the advent of low cost sensors, and advances in wireless and broad band technology there is an increasing interest in software agents as a paradigm for modeling and developing intelligent monitoring applications among others, for ambient assisted living and wellness, entertainment, logistics, energy management and industrial automation [23]. Continuous observations from the physical systems being monitored may contain new patterns which, when followed up can lead to knowledge discovery and evolution (KDE) [12].

While KDE is routinely performed by humans, formalising and automating the KDE process is difficult. In Philosophy of Science (PoS), KDE focuses on theories and methods that scientists apply in theory discovery and justification. With prominent theories of scientific discovery like the inductive theory and the hypothetico-deductive (HD) theory of method, philosophers of science have sought to lay out the discovery process in an orderly manner. The inductive theory focuses on creating and justifying theories by discovering empirical generalizations in the data while in HD, the scientist acquires a hypothesis and aims

⋆ Supported by Hasso Plattner Institute (HPI) for Digital Engineering

to test it's predictive success [10]. A more recent abductive theory of method (ATOM) [10, 11] consists of two overarching processes i.e. *phenomenon detection* in which phenomena (unexplained "relatively stable, recurrent, general features that researchers aim to explain in the data") are detected and *theory construction* whereby possible explanations are provided for the detected phenomena. The phenomenon detection process consists of four activities i.e. initial data analysis, exploratory data analysis, close replication and constructive replication which in our work translate to preprocessing, model building, pattern recognition and confirmation. Theory construction encompasses three major activities; theory generation, theory development and theory appraisal.

ATOM provides a more encompassing theory of method and deals with some of the limitations of the HD and inductive methods. It provides a concrete approach for formulating and generating theories and it provides explanations for the obtained empirical generalization. ATOM accommodates both top down and bottom up AI techniques [32] and, while it emanates from the behavioural sciences, it is applicable to a broad array of complex social, physical and socio-technical systems, especially for KDE in intelligent monitoring applications.

In this paper, we explore and propose a generic architecture for a KDE agent inspired by ATOM. We use an example of a real world data driven sensor application for monitoring and understanding daily electricity consumption behaviour in households across South Africa [29–31] to design and analyse the architecture.

We identify two key actors in the agent's environment i.e. an external AI service and the domain expert. The agent uses the AI service for pattern and anomaly detection and relies on the domain expert for knowledge acquisition and theory appraisal. Internally, the agent incorporates a combination of ontologies, rules and Bayesian networks to represent knowledge and generate explanations for unexpected patterns. The architecture shows how different AI techniques can be used to deal with situation detection and analysis, generating explanations and updating the agents beliefs to reflect changes in the physical system.

The rest of the paper is organised as follows: Sect. 2 presents the background and related work. Section 3 presents the design of the architecture. In Sect. 4, we discuss the proposed KDE architecture and present an evaluation of the architecture by demonstrating its applicability in the domestic electricity consumption domain in Sect. 5. We then discuss and conclude in Sect. 6.

## 2 Background and related work

### 2.1 The BDI architecture

The BDI architecture is one of the most prominent agent architectures [21, 22]. It views an agent as consisting of three mental attitudes i.e. beliefs, desires and intention. Beliefs consist of the agent's knowledge about its environment, itself and other agents. Desires motivate the agent since they represent its objectives or goals. Intentions represent a subset of desires that the agent has committed to achieving. At run-time, in response to the percepts, the BDI interpreter updates the agent's beliefs and goals and manages its intentions [3].

The BDI language is advantageous for agent development because it allows "rapid development, context sensitive and robust behavior, intelligibility and verifiability due to its descriptive and intuitive nature, it facilitates a range of symbolic, stochastic and sub-symbolic AI techniques", formalises the reasoning process in a straightforward way, facilitates "knowledge capture, representation and debugging" and allows the exploitation of available qualitative information, commonsense evidence and psychological theories in the modelling process [1, 3]. However, the BDI architecture was originally designed for practical reasoning in real-time planning applications [4]. In its original form, it may not be able to support a KDE agent in more complex data-driven physical systems. The architecture does not provide specific support for processing and detecting patterns from large amounts of data. While it supports mechanisms for deliberation and decision making, generating explanations for non-technical human users is not a primary objective. Supplementing typical BDI agents with machine learning services can deal with pattern recognition. While, at first glance one may be inclined to abstract this out and consider it to be just an implementation or engineering issue, which is internal to the agent's perception and deliberation modules, not explicitly considering this can affect its adoption and usage for implementing real world intelligent monitoring systems.

### 2.2 Integrating AI into BDI agents

BDI agents can be enriched by drawing from and incorporating a variety of AI techniques in a number of ways. Bordini et al., [3] discuss the applicability of AI in the sensing, planning and acting phases of BDI agents e.g. in the sensing phase, AI can be used to enrich and update the underlying knowledge representation to handle complex ontologies [17, 19] or to work with probabilistic knowledge, beliefs and goals using Bayesian networks [25]. Bordini et al., also present some approaches for integrating AI into the BDI agent architecture i.e. i) AI as a service, an exogenous approach where the agent exploits AI e.g. external image/speech recognition and document analysis. ii) AI embedded into agents, an endogenous approach in which AI components replace or complement the standard elements of the BDI architecture or cycle e.g the use of Bayesian networks in [7]. iii) hybrid approach in which some AI components are endogenous and others are exogenous. Complementing agents with AI services would leverage complex AI algorithms providing advanced analytical processing, making predictions and supporting knowledge discovery from data [18]. Some architectures are discussed in [18] in which the agent and the AI service can co-exist and cooperate to deliver decision support, one of which (shown in Fig. 1) caught our attention because it can potentially be applied in a KDE agent. In a KDE task environment, different AI algorithms can be applied as part of external AI services to enable the adaptability of the agent by facilitating the understanding of the behavior of data acquired from the operation domain as well as the detection of new patterns in new observations. In [7] on the other hand, an endogenous approach is followed where Bayesian networks are used internally to abstract the BDI mental states and to deliberate about and select the optimal action.

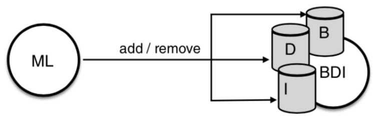


**Fig. 1.** The AI service (ML component) influences latter agent activities e.g. by adding plans or beliefs that could be synthesised from the agent's operation domain [18].

### 2.3 KDE systems and approaches

KDE encompasses pattern detection in observations, situation detection and understanding or explaining their occurrence in order to update or improve the body of knowledge in the respective domain. Patterns, which result from a data mining step [28] are a subset of data behaving the same way [27].

A recent study done to design an agent architecture for novelty detection, characterisation and accommodation in a changing world [20] presents a promising approach when agents encounter sudden changes in their environments. However, it is limited when the agent's role is to detect novel situations in observations with the goal of explaining them. In [24], a domain-specific architecture that combines ML with automated reasoning is applied in taxonomic intelligence i.e. discovering patterns of taxonomic identity and change which involves learning from biodiversity data and reasoning over existing biodiversity knowledge.

Some systems have been designed for KDE e.g. the Robot scientists [14–16] where the aim is to implement laboratory-automated systems with the help of AI techniques to execute cycles of scientific experiments, and the DISK(Automated DIscovery of Scientific Knowledge) system [8, 9] that aims at achieving automated hypothesis driven data analysis of science data repositories. Coetzer et al., [5] also developed a domain specific knowledge-based system for discovering ecological interactions in data in which situation detection is performed by combining the observations with relevant available domain knowledge.

Context modelling and situation awareness call for solutions that are semantically designed or incorporate ontologies [26]. Some monitoring systems for sensor based applications have been described for instance in [2] in which air quality monitoring is used as an example application use case. Although in [2] focus is not on KDE but on proactively making decisions and acting on anticipated situations with regard to a continuously monitored property of a feature of interest, we draw ideas from the situation detection layer of this system architecture. In [13] and [26], rule engines i.e. semantic web rule language (SWRL) rules are used. These studies, together with [2] provide approaches for semantic/ontology driven monitoring to generate alerts. The rules are used for situation detection, alerting and transforming continuous variables to qualitative measures. As noted in [6], the main shortcoming of most of the available systems is lack of continuous knowledge base update or knowledge evolution. However, some systems have been designed to overcome this shortfall e.g. the ontology driven health monitoring system, Do-Care [6] proposed recently in the medical domain that allows update and refinement of medical knowledge triggered by observations with regard to wearable, nearable and usable devices. This system and others like the ones described in [13] and [18] focus on generating appropriate alerts when wrong, risky or abnormal observations are encountered as opposed to providing potential explanations for unexpected situations.

While there are elements that can be reused in the systems and approaches discussed in this section, an explicit generic architecture for agent-based KDE is absent. We can incorporate the ideas, experiences, lessons learned and the techniques applied in these systems into the design of such an architecture.

## 3 Design of the KDE agent architecture

To inform our design we considered a real world application use case for monitoring electricity consumption behavior in South African households [31]. A simplified summary of the application is given below.

Daily consumption behaviour is highly variable and may differ drastically between households, due to economic volatility, income inequality, geographic and social diversity. Energy planners must understand the electricity consumption behaviour of residential customers in order to predict long term energy demand. The aggregate consumption behaviour, or representative load profiles, of residential customers are standard consumption patterns for dominant groups of households that have common attributes. These representative load profiles consolidate expert knowledge and represent the electricity consumption of typical customer classes. They are an essential tool for demand planning, but are difficult and tedious to construct and do not cater for changes in household behaviour. This is a serious limitation that impacts energy demand planning. Through clustering, the daily electricity consumption behavior and representative daily load profiles are already known [29]. However, the consumption patterns, household profiles and the original clustering are not static and may become obsolete as household characteristics evolve or new groups emerge.

A KDE agent in this environment would interact with an external cluster analysis service and the energy planner. The cluster analysis service provides the current representative load profiles for different types of households and the agent acquires knowledge from the energy planner who serves as the domain expert. The main task of the agent is to maintain and update knowledge about daily consumption behaviour. It monitors individual household consumption on a daily basis, detects and tracks abnormal consumption, i.e. when a household deviates from its expected pattern, and detects new consumption patterns, provides plausible explanations for this, and interacts with the energy planner to updates its knowledge.

The agent architecture is governed by the following design goals/principles.

1. The agent must be able to interpret and analyse incoming observations from the physical system, in this case electricity consumption in a household, via an external AI service
2. The agent must be able to interact with the domain expert, in this case the energy planner, to acquire and align with the domain expert's knowledge of

the observed physical systems, and must be able to communicate adequate explanations to the domain expert.
3. The agent must be able to represent and align with the different steps for KDE specified by ATOM.
4. The agent architecture must be generic and support a general class of intelligent monitoring systems for sensor based applications.

We used the mental states specified in the BDI architecture to analyse the application use case and to develop the cognitive loop of the KDE agent. The agent's role is to generate, develop and partly appraise explanations and theories that explain a given pattern and to update its domain knowledge. It has a single goal and intention, i.e. to provide explanations for patterns in incoming observations.

The agent's cognitive loop consists of the following steps; detect the current situation, determine if the detected situation is expected or unexpected, generate plausible explanations for the detected situation, revise beliefs and share explanations and theories with the human experts. On obtaining a pattern, the KDE agent's intention is $achieve(explained\ ?x)$ where $x$ is the detected pattern. The agent maintains a set of beliefs which represent its knowledge. It relies on its beliefs to deliberate and generate possible explanations for the observed situations. From these potential explanations, the agent selects the best explanation which is to be followed up further by the human expert.

## 4 The KDE agent architecture

Fig. 2 shows the layout of the proposed architecture which applies a hybrid approach where AI is applied both endogenously and exogenously. The architecture is presented as modules connected through data and control flow. It consists of five main distinct and yet dependent modules. These are: The AI service, perception module, deliberation module, the theory construction module and the domain expert module. Details about these modules are discussed below.

### 4.1 The exogenous modules

The exogenous components comprise of two modules; the AI service and the domain expert.

**The AI service:** As part of the AI service, bottom up techniques like Machine learning and deep learning can be used to build models from data in the domain (arrow 1). Clustering techniques, image or speech recognition systems, document analysis capabilities, etc, can be offered as part of the AI service component. The models are used to represent the expected behaviour e.g. the clustering that is used in section 5.1 to represent the household electricity consumption behaviour. The pattern detection (PD) service matches the current behaviour of the observed entity with the expected behaviour and reports when there is any deviation.

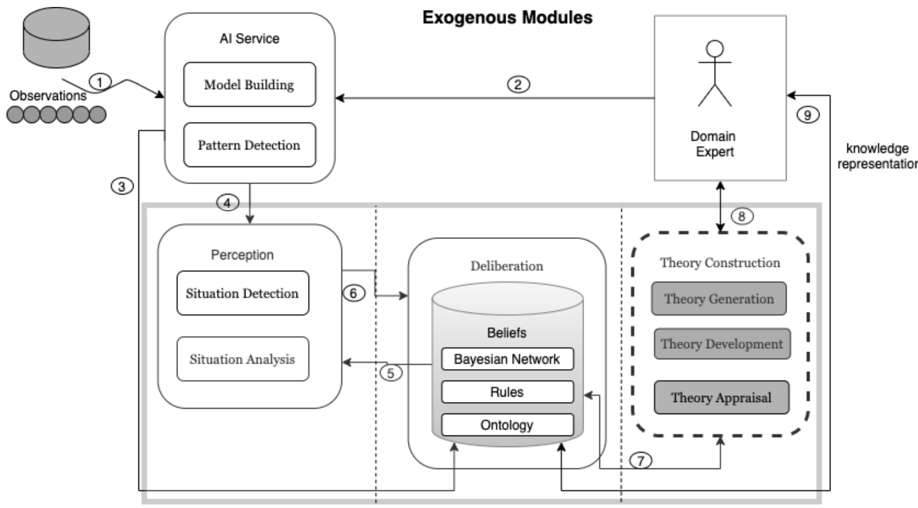


**Fig. 2.** The KDE abstract agent architecture.

**The domain expert:** The domain experts has two overarching roles. They facilitate the bottom up process of model building (arrow 2). They also give input to the top down knowledge representation and reasoning component used to capture the agent's beliefs and known domain theories i.e. Bayesian networks, rules and ontologies (arrow 9) as well as retrieving information from them. While the Bayesian network typically represents causal domain knowledge acquired from the expert it can also be learned or refined from the data (arrow 3).

### 4.2 The endogenous modules

The modules that are endogenous to the agent include the perception, deliberation and theory construction modules.

**The perception module:** The agent receives as its percepts the patterns that result from the pattern detection component of the AI service (arrow 4). The agent has rules i.e. SWRL rules as part of its beliefs that are used to analyse the incoming patterns in order to determine the current situation that the observed pattern represents (arrow 5). The rules are captured using the template:
`Pattern` $\Rightarrow$ `Situation`.

As far as the agent is concerned, two different types of patterns may be detected from the observations. These include: expected and unexpected patterns. Expected patterns represent situations that are already defined and explained by current model. Unexpected patterns on the other hand are new to the agent and in some cases to the domain expert. They are unexplained patterns that represent situations that have not been seen previously. The agent then has to carry out situation analysis to establish whether the situation that the pattern is indicative of is expected or unexpected.

**Deliberation:** The agent attempts to diagnostically generate possible causal explanations (arrow 6) for the detected situations. The assumption is that unlike unexpected situations, expected situations are already correctly explainable by the agent based on its current beliefs/knowledge. The base component of the agent's deliberation process are its beliefs. The agent needs to know the domain variables, their characteristics and their causal interaction in order to be able to generate tentative explanations as to why a particular situation has occurred. In the proposed architecture, beliefs are maintained using ontologies, rules and Bayesian networks. The Bayesian network consists of domain theories captured as causal relationships between domain variables from which explanations about the occurrence of a particular situation are obtained. The ontology is used to integrate aspects of generated explanations or theories. The integration process is dictated by the aspects captured as part of the KDE ontology [32] which is linked to the agent's beliefs. This task entails combining/linking aspects about the patterns, situations and the generated theories including any other required provenance information that come from the discovery process. The KDE ontology captures knowledge in a queryable way which makes it possible for the humans to query and retrieve the KDE information about the generated explanations or theories and their provenance information and further appraise the theories generated by the agent. Parameterised SPARQL queries are used for this aspect in order to query the KDE ontology.

*Generating explanations:* Explaining a given situation commences when the Bayesian network is set to capture the detected situation (Algorithm 1 line 4). An abstract Bayesian network that lays out the the aspects of Algorithm 1 is shown in Fig 4 appendix 1. Obtaining plausible causal explanations is done through diagnostic reasoning. Causality is leveraged to determine the tentative explanations of a detected situation. Algorithm 1 lays out the procedure of tentative explanation generation from the agent's beliefs in which the intention is to explain the detected situation. Probabilistic information is used by the agent to determine whether a particular explanation is worth further exploration. The assumption made is that if a variable has a causal link to the situation node, then it can tentatively explain the occurrence of the observed situation. All nodes that are parents of the situation node in the Bayesian network are determined and their states that have the highest probabilities are obtained and compared with the predetermined threshold value. The nodes along with the states that present with higher probabilities than the rest of the nodes are added to the possible explanations. (see Algorithm 1 line 5-8).

When probabilities are propagated and the probabilities of the rest of the nodes in the Bayesian network are updated to incorporate the detected situation, states of nodes that have probabilities that are higher than a predetermined probability threshold translate to possible explanations. The possible explanations are further compared in terms of their probabilities for explanatory strength in order to select the best explanation. The theory appraisal task in our architecture is designed to be started by the agent and completed by the human scientist or domain expert. The aspect of theory appraisal that is carried out by the agent entails obtaining the best explanation for the observed situations.

**Algorithm 1:** Selecting possible explanations

options: (B,I,S) $\rightarrow$ *possibleExplanations*
    E := null;
    X := nodes in the Bayesian Network; /* *the Bayesian network is part of the Beliefs*
    Set situation nodes in X to match the situation S
    **for each** $x \in$ *Parents(S)* **do**
        **if** max $(P[x = x_i|S = s])$ **then**
            **if** $P$ >= beliefThreshold **then**
                add $(x, x_i)$ to $E$
            **end-if**
        **end-if**
    **end-for**
    options := E;
end-options

**Theory construction:** Theory construction consists of three major processes which are theory generation, theory development and theory appraisal which are facilitated by the domain expert. Theories could be constructed for a new pattern or possible patterns. This work mainly focused on explaining encountered patterns with the help of already existing theories.

## 5 Use Case - Domestic Electricity Consumption

This application use case is based on an existing study which used cluster analysis to determine prominent patterns of daily electricity consumption by different households in South Africa [31]. The daily consumption pattern for a household, $h$, is reflected by a daily load profile. The load profile, $h_i$, is a 24 element array of average consumption measurements for each hour in day $i$. Households are characterised by different socio-economic attributes, e.g. income, years of electrification, ownership of high power appliances and whether there is piped water and other complementary infrastructure. Consumption patterns vary depending on the season, the day of the week and the time of day. For example for a typical week day, households will consume more electricity in winter because of heating, and more electricity in the mornings and evenings, which are peak times when families are at home, meals are being prepared and appliances are used.

Daily consumption behaviour can vary drastically for individual households over time. The dominant daily consumption patterns can become outdated as new groups of households emerge which may not correspond to the current patterns. An example of this is households in rural areas in South Africa, where thatch roof huts with limited appliances have gradually transitioned to brick buildings with modern appliances, resulting in a significant change in electricity consumption. Another issue is that while daily consumption readings are always available for a given period, socio-economic attributes are collected by a once off survey. This data will eventually become outdated as the household changes and evolves and may eventually not align with the observed daily consumption for a household with the attributes on record.

Within this context we consider an individual household $h$ that is being monitored for its daily electricity load consumption.

### 5.1 Cluster analysis service

The cluster analysis service contains the cluster set which represents the prominent daily consumption patterns (daily load profiles) for all households in South Africa. The reader is referred to [31] for details on how such a cluster set can be generated. A cluster represents daily load profiles which are similar with the cluster centroid representing the mean daily load profile for profiles in the cluster. For this household the cluster analysis service can provide the cluster which best represents the consumption for this household based on its socio-economic attributes and daily consumption pattern in a given season and time of day. The pattern detection (PD) service builds a model of the expected usage over time for this household. There are different usage patterns for different seasons (winter, summer) and time of day (morning, evening). This model could be a simple average of previous daily load profiles from historical data for the household or a more complex machine learning model which can predict the load profile for a given season. Given the current load profile, the PD service will match the current daily consumption with the expected consumption and return whether it aligns with the expected consumption.

### 5.2 Perception

The agent uses its rule set to analyse the current daily consumption pattern. We provide a restricted rule set (example rules are listed below and are presented as SWRL rules in appendix 2) to analyse the consumption, $c$, on a week day morning between 5am and 8am in summer.

`day_type(summer_week_day) ∧ time_of_day(morning) ∧ usage(?x>74) ⇒ high_morning_consumption_summer_week_day`

`day_type(summer_week_day) ∧ time_of_day(morning) ∧ usage(49<?x<75) ⇒ medium_morning_consumption_summer_week_day`

`day_type(summer_week_day) ∧ time_of_day(morning) ∧ usage(?x<50) ⇒ Low_morning_consumption_summer_week_day`

The agent uses its rule set to analyse the expected morning consumption pattern which it expects to be *medium_morning_consumption_summer_week_day* but the current pattern yields that it is a *high_morning_consumption_summer_week_day*

### 5.3 Deliberation to generate explanations

The agent then attempts to provide explanations for this new behavior with the help of a causal Bayesian network that is part of the agent's beliefs. Consider a simple example Bayesian network for electricity consumption characteristics in Fig. 3. (The Bayesian network and conditional probability tables (CPTs) do not depict a true setting but it has been simplified to illustrate a concept). The Bayesian network presents characteristics of houses that have the same consumption behaviour. The agent then attempts to explain why $h$ might be exhibiting a high consumption behavior by setting the network, in the situation nodes, to accommodate a high consumption in the morning on a summer weekday. The agent determines all the nodes that have causal links to the situation node i.e. complementary infrastructure e.g piped water, high consumption appliance ownership and years of electrification. The states of these nodes with higher probabilities are selected i.e complementary infrastructure e.g. piped water (yes) -52.5%, high consumption appliance ownership (yes) - 73.5% and >=15 years of electrification 42.3%. Assuming the threshold is 50%, complementary infrastructure e.g piped water (yes) -52.5%, and high consumption appliance ownership (yes) - 73.5% are added to the possible explanations and high consumption appliance ownership is ranked best. The generated explanation and its provenance information are then inserted into and retrieved from the the agent's KDE ontology using SPARQL queries. We argue that the agent's deliberation to generate explanations constitutes a weak form of analogical reasoning as described by Haig [10]. Households with high consumption behavior in the morning on weekdays in summer are households with complementary infrastructure or high consumption appliances. Using analogical reasoning we infer that this household could be one of these households.

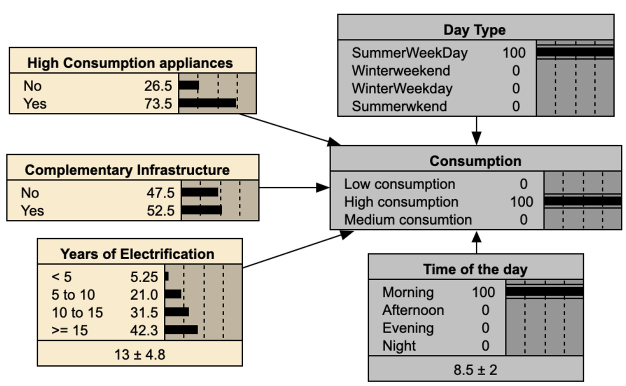


**Fig. 3.** A sample Bayesian network.

## 6 Discussion and Conclusion

We have proposed an agent architecture for knowledge discovery and evolution in dynamic and erratic physical sensor systems. We draw from and attempt to formalise the knowledge discovery and evolution process presented in ATOM. We use the BDI architecture as a cognitive model for the agent and hone in on the maintenance and evolution of the agent's beliefs. In this scenario the agent has a single persistent goal to explain its observations in order to evolve and align its beliefs with current observations in the system. Its beliefs must be in a form that can offer explanation and and allow for assessment by a domain expert. At the end of its deliberation process, the agent's goal is to answer the question: "what do I think is the best explanation for the observed percept?". The architecture builds on both the mental states from the BDI and the agent community and the ATOM process of KDE from PoS. We also show how external machine learning services and the domain expert can be incorporated into the agent's deliberation. We believe that an agent's beliefs will comprise of rules, Bayesian networks and ontologies. While many data driven sensor applications have been proposed which use a combination of rules and ontologies, they have not applied these causal theories and reasoning about uncertainty. We show how Bayesian networks can be incorporated into an agent for representing causal theories and applying these theories to identify possible explanations for unexpected situations.

The use case application illustrates certain practical aspects of the architecture for detecting and analysing unexpected observations, finding possible explanations for unexpected situations and establishing how the agent can constantly align its beliefs to changes in the environment.

Exploring new mechanisms for KDE is essential for the next generation of intelligent agents. To our knowledge this is the first attempt to formalise ATOM into an intelligent agent architecture for situation analysis and explanation generation in a physical data driven sensor application. The proposed architecture applies AI techniques for the detection of unexpected situations and generating possible explanations for those situations. The theory construction module requires further exploration. We illustrated a weak form of analogical modeling for theory development as proposed by Haig [10], but will explore richer forms of analogical modeling in future work. We also intend to explore the applicability of the proposed architecture to other use cases to evaluate its generalisability for the broader class of sensor based applications.

## 7 Acknowledgements

This work was financially supported by the Hasso Plattner Institute for digital engineering through the HPI Research school at UCT.

## References


1. Adam, C., Gaudou, B.: Bdi agents in social simulations: a survey. Knowledge Engineering Review **31**(3), 207–238 (2016)
2. Adeleke, J.A.: A semantic sensor web framework for proactive environmental monitoring and control. Ph.D. thesis, University of KwaZulu-Natal, Durban (ukzn) (2017)
3. Bordini, R.H., El Fallah Seghrouchni, A., Hindriks, K., Logan, B., Ricci, A.: Agent programming in the cognitive era. Autonomous Agents and Multi-Agent Systems **34**, 1–31 (2020)
4. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Computational intelligence **4**(3), 349–355 (1988)
5. Coetzer, W., Moodley, D., Gerber, A.: A knowledge-based system for generating interaction networks from ecological data. Data & Knowledge Engineering **112**, 55–78 (2017)
6. Elhadj, H.B., Sallabi, F., Henaien, A., Chaari, L., Shuaib, K., Al Thawadi, M.: Do-care: A dynamic ontology reasoning based healthcare monitoring system. Future Generation Computer Systems **118**, 417–431 (2021)
7. Fagundes, M.S., Vicari, R.M., Coelho, H.: Deliberation process in a bdi model with bayesian networks. In: Pacific Rim International Conference on Multi-Agents. pp. 207–218. Springer (2007)
8. Gil, Y., Garijo, D., Ratnakar, V., Mayani, R., Adusumilli, R., Boyce, H., Mallick, P.: Automated hypothesis testing with large scientific data repositories. In: Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems (ACS). vol. 2, p. 4 (2016)
9. Gil, Y., Garijo, D., Ratnakar, V., Mayani, R., Adusumilli, R., Boyce, H., Srivastava, A., Mallick, P.: Towards continuous scientific data analysis and hypothesis evolution. In: AAAI. pp. 4406–4414 (2017)
10. Haig, B.D.: An abductive theory of scientific method. In: Method matters in psychology, pp. 35–64. Springer (2018)
11. Haig, B.D.: The importance of scientific method for psychological science. Psychology, Crime & Law **25**(6), 527–541 (2019)
12. Han, J., Kamber, M., Pei, J.: Data mining concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems **5**(4), 83–124 (2011)
13. Hristoskova, A., Sakkalis, V., Zacharioudakis, G., Tsiknakis, M., De Turck, F.: Ontology-driven monitoring of patient's vital signs enabling personalized medical detection and alert. Sensors **14**(1), 1598–1628 (2014)
14. King, R.D., Rowland, J., Aubrey, W., Liakata, M., Markham, M., Soldatova, L.N., Whelan, K.E., Clare, A., Young, M., Sparkes, A., et al.: The robot scientist adam. Computer **42**(8), 46–54 (2009)
15. King, R.D., Rowland, J., Oliver, S.G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L.N., et al.: The automation of science. Science **324**(5923), 85–89 (2009)
16. King, R.D., Whelan, K.E., Jones, F.M., Reiser, P.G., Bryant, C.H., Muggleton, S.H., Kell, D.B., Oliver, S.G.: Functional genomic hypothesis generation and experimentation by a robot scientist. Nature **427**(6971), 247–252 (2004)
17. Klapiscak, T., Bordini, R.H.: Jasdl: A practical programming approach combining agent and semantic web technologies. In: International Workshop on Declarative Agent Languages and Technologies. pp. 91–110. Springer (2008)

18. Montagna, S., Mariani, S., Gamberini, E., Ricci, A., Zambonelli, F.: Complementing agents with cognitive services: A case study in healthcare. Journal of Medical Systems **44**(10), 1–10 (2020)
19. Moreira, A.F., Vieira, R., Bordini, R.H., Hübner, J.F.: Agent-oriented programming with underlying ontological reasoning. In: International Workshop on Declarative Agent Languages and Technologies. pp. 155–170. Springer (2005)
20. Muhammad, F., Sarathy, V., Tatiya, G., Goel, S., Gyawali, S., Guaman, M., Sinapov, J., Scheutz, M.: A novelty-centric agent architecture for changing worlds. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. pp. 925–933 (2021)
21. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a bdi-architecture. KR **91**, 473–484 (1991)
22. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: from theory to practice. In: Icmas. vol. 95, pp. 312–319 (1995)
23. Savaglio, C., Ganzha, M., Paprzycki, M., Bădică, C., Ivanović, M., Fortino, G.: Agent-based internet of things: State-of-the-art and research challenges. Future Generation Computer Systems **102**, 1038–1053 (2020)
24. Sen, A., Sterner, B., Franz, N., Powel, C., Upham, N.: Combining machine learning & reasoning for biodiversity data intelligence. Proceedings of the AAAI Conference on Artificial Intelligence **35**(17), 14911–14919 (2021)
25. Silva, D.G., Gluz, J.C.: Agentspeak (pl): A new programming language for bdi agents with integrated bayesian network model. In: 2011 international conference on information science and applications. pp. 1–7. IEEE (2011)
26. Sondes, T., Elhadj, H.B., Chaari, L.: An ontology-based healthcare monitoring system in the internet of things. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). pp. 319–324. IEEE (2019)
27. Tiddi, I., d'Aquin, M., Motta, E.: An ontology design pattern to define explanations. In: Proceedings of the 8th International Conference on Knowledge Capture. pp. 1–8 (2015)
28. Tiddi, I., d'Aquin, M., Motta, E.: Dedalo: Looking for clusters explanations in a labyrinth of linked data. In: European Semantic Web Conference. pp. 333–348. Springer (2014)
29. Toussaint, W.: Evaluation of clustering techniques for generating household energy consumption patterns in a developing country. Master's thesis, Faculty of Science, University of Cape Town (2019)
30. Toussaint, W., Moodley, D.: Comparison of clustering techniques for residential load profiles in south africa. In: Davel, M.H., Barnard, E. (eds.) Proceedings of the South African Forum for Artificial Intelligence Research Cape Town, South Africa, 4-6 December, 2019. CEUR Workshop Proceedings, vol. 2540, pp. 117–132. CEUR-WS.org (2019)
31. Toussaint, W., Moodley, D.: Automating cluster analysis to generate customer archetypes for residential energy consumers in south africa. arXiv preprint arXiv:2006.07197 (2020)
32. Wanyana, T., Moodley, D., Meyer, T.: An ontology for supporting knowledge discovery and evolution. In: Gerber, A. (ed.) Southern African Conference for Artificial Intelligence Research (SACAIR). pp. 206–221 (2020)

# 8 Appendices

## 8.1 Appendix 1 -An abstract Bayesian network

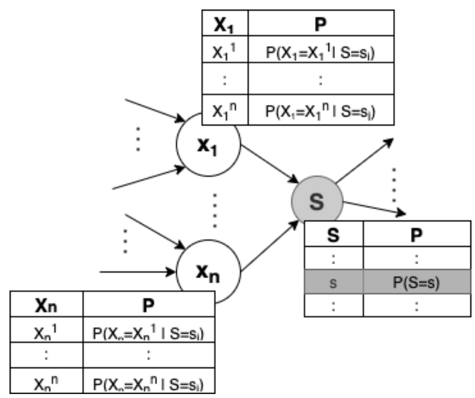

**Fig. 4.** An abstract Bayesian network.

## 8.2 Appendix 2 - Sample SWRL rules

```
Kdeontology:Household(?h) ∧ Kdeontology:has_Consumption(?h, ?c)
∧ Kdeontology:Day_type(Kdeontology:summer_week_day)
∧ Kdeontology:Time_of_day(Kdeontology:morning)
∧ swrlb:greaterThan(?c, 74) → Kdeontology:has_situation(
?h, Kdeontology:high_morning_consumption_summer_week_day)

   Kdeontology:Household(?h) ∧ Kdeontology:has_Consumption(?h, ?c)
∧ Kdeontology:Day_type(Kdeontology:summer_week_day)
∧ Kdeontology:Time_of_day(Kdeontology:morning)
∧ swrlb:greaterThan(?c, 49) ∧ swrlb:lessThan(?c, 75)
→ Kdeontology:has_situation(
?h, Kdeontology:medium_morning_consumption_summer_week_day)

   Kdeontology:Household(?h) ∧ Kdeontology:has_Consumption(?h, ?c)
∧ Kdeontology:Day_type(Kdeontology:summer_week_day)
∧ Kdeontology:Time_of_day(Kdeontology:morning)
∧ swrlb:lessThan(?c, 50) → Kdeontology:has_situation(
?h, Kdeontology:low_morning_consumption_summer_week_day)
```