DEVELOPMENT OF A PRIVACY IMPACT ASSESSMENT ONTOLOGY FOR THE INTERNET OF THINGS

Kent O'Sullivan - u1092557

Feb 2022

Supervisor: Associate Professor Xiaohui Tao

A thesis submitted in partial fulfilment of the requirements of the degree Master of Data Science

All code is available on the Author's Github

Abstract

The intersection of the Internet of Things and Privacy domains has created new, unprecedented complexity. There are substantial knowledge gaps around how to estimate the impact of IoT devices on the privacy of an individual. Semantic data structures are a promising prospect, but are hampered by their cumbersome size, domain specificity and very manual development approaches. This thesis describes the application of agile ontology development approaches to develop the Privacy Impact Assessment Nexus Ontology (PIANO) and the Social Internet of Things Knowledgebase. The development of the knowledgebase employs semi-automated ontology learning to derive new knowledge about the intersection of IoT devices and an individual's privacy. The development and evaluation use-case is derived from the Social Internet of Things project dataset, and its application reveals the significant potential of PIANO, and importantly the development methodology that is driving its development to finally bridging the gap between IoT and Personal Privacy. The conditions are set for future work in this area, linking the abstract PIANO structure to granular IoT and Personal ontologies, demonstrating its utility outside of a pure proof-of-concept simulation

1 Thesis certification page

This Thesis is entirely the work of Kent Daniel O'Sullivan except where otherwise acknowledged. The work is original and has not previously been submitted for any other award, except where acknowledged.

Principal Supervisor: Associate Professor Xiaohui Tao

2 Acknowledgements

I wish to publicly thank my thesis supervisor, Associate Professor Xiaohui Tao for his guidance and support throughout this project. I certainly was not the model student throughout this project and I'm eternally grateful for your patience and support.

Thanks also go to The University of Southern Queensland who throughout my entire Masters program have been tolerant and flexible, working with me to accommodate the varying demands of studying while providing full-time military service. I'd recommend studying with USQ to any other military member based on my experience since 2016.

Final thanks go to my partner Mary, whose patience and support over months of thesis-related stress hasn't gone unnoticed. I'm sorry for taking over your study for so long. Keep helping me to be better in everything I do. I promise I'll finally buy a bike now, that I don't have writing this thesis as an excuse.

Contents

3 Introduction

3.1 Background

The *Internet of Things* can be thought of as the interconnection of objects or *things* in the physical world and their representations on the internet[1]. Those *things percieve* the physical world, communicate via some kind of *network* and feed into some kind of *application* [2]. The advent of the internet of things is pushing the development of an increasingly connected, collective environment. Working to overcome challenges of scale, heterogeneity, unknowable network topologies, incomplete metadata and device conflict resolution [3] has driven the development of semantic data structures to facilitate interoperability of IoT devices [3], [4].

Ontologies are, in computer science terms, semantic data structures that serve as the explicit specification of a shared conceptualisation [5]. They are instrumental when it is our goal to combine multiple interpretations or points of view into a single, intelligible, shared definition [6], such as, for example, when combating heterogeneity. Ontologies as semantic structures can serve as a kind of *middleware* to enable interoperability. While underlying best practice for ontology development is well established [7]–[9], an emerging requirement for the IoT domain is balancing expressiveness and utility[1]. Recently the focus has pushed much more towards lightweight ontologies that support rapid, low-cost processing. With the advent of 5G, the IoT devices will become smaller, cheaper and more pervasive as a reliable high bandwidth network backhaul [10] allows almost all processing to be done remotely. The proliferation of IoT devices further into peoples homes, workplaces and, in some cases, bodies raises urgent questions about the impact the devices could have on privacy.

Privacy is a tricky topic that refuses to be clearly defined [11]. The most accepted approaches have expressed the harms or principles of privacy as frameworks, typologies or taxonomies[11]–[13]. However, the work towards developing privacy ontologies has been driven by niche domain use cases or by imposed regulation specific to only subsets of IoT users[14]–[16]. From an ontological perspective, the consequence is that the ability to understand the risks to privacy posed by IoT devices cannot be answered with any of the extant work in that area.

Both the domains of IoT and privacy are complicated and dynamic. Manual approaches to designing, updating and maintaining static ontologies have proved to be effective if time-consuming in many fields [8], [17]–[22]. However, manual approaches will continue to be a limiting factor to the usability and relevance of ontologies in complex, dynamic domains like Privacy and IoT. If they are unmaintained, the concepts and relations in them will not reflect reality. Further, they will not be capable of providing the expected answers to questions of

inference. Ontology learning offers a path to try and overcome the limitations of manual ontology development[23], [24]. However, substantial work needs to occur to understand how to apply ontology learning to semi-structured technical artefacts and how to ensure that the knowledge of privacy is either broad enough to be applicable anywhere or flexible enough to be trained to specificity.

4 Aims

The **objective** of this thesis is to produce a *methodology* that will make a positive contribution to the application of ontology learning in the privacy and internet of things domains. The robustness of this contribution will be assured through formal methodology derived from best practice in the field. The use of a formal approach will structure the research, inform the lines of inquiry and enable the ongoing scoping of the problem throughout the research process.

4.1 Business Problem

Widely accepted approaches to data science and data mining, in particular, emphasise the importance of first specifying the business problem that needs to be solved [25]. Succinctly, the problem can is:

> The proliferation of IoT devices into the modern world and the looming adoption of 5G creates a paradigm where individuals do not understand the risk to their physical or data privacy. They have lost the ability to clearly understand and make decisions about how their data is used, by whom and for what purpose. To regain control of their privacy, individuals should be able to apply personalised privacy preservation protocols that specify the controls they want to be enforced on the IoT devices they encounter. For personalised privacy protocols to be applied, a recommender system has to be able to 'understand' the risk posed to the individual's privacy based on their personal preferences and the capabilities of the IoT devices they are exposed to.
>
> There is currently no mechanism available to estimate the privacy implications of IoT devices.

4.2 Scientific Problem

Generally, the second step in approaching data science problems to gain an understanding of the data [25]. Preliminary analysis has identified two publicly

available datasets. The *University of New South Wales* **TON-IoT Dataset** [26], and the *Social Internet of Things* **Network IoT Dataset** [27]. Both datasets consist primarily of comma-separated value (CSV) files, excel spreadsheets and plain text files, which contain log data from various IoT devices. Effectively, the scientific problem faced here is:

> *How can we derive an ontological structure that enables privacy impact assessment from IoT device data.*

Or, more specifically for this research project:

> *What is an appropriate methodology to derive ontological structures taht enable privacy impact assessments from IoT device data.*

4.3 Research Question

The proposed research question, based on the literature review conducted, is:

1. What is an ontology development methodology that can dynamically generate ontological structures to support privacy impact estimation for the internet of things.
   (a) What is a suitable ontology development method to derive semantic relationships from semi-structured source data such as XML files, PCAP files and log files?
   (b) What is a suitable ontology development method to be used to derive semantic links between IoT device data and privacy concepts?
   (c) What is a suitable system architecture to enable the conduct of these two ontology learning tasks?

4.4 Research Scope

The intended path of this research is employing a *design science research methodology* (DSRM) to create a deliverable artifact that demonstrates the viability of automatically or semi-automatically generating ontological structures. A system use case will be used to achieve the intent of a DSRM approach. However, it will be *limited* in the scope of implementation. Many aspects of the use case, particularly system interfaces and the actual development of any ontological structure, will be abstracted with stubs or mock data. Further, this work will not examine how to automate the generation of privacy ontologies, using an ontological stub instead to demonstrate the interface.

The intended development approach is based on the agile philosophy and will build the system's minimally viable core. Subsequent iterations will add new features as time permits further development occurring.

Where no best practice is found in research (for example, regarding privacy policies), the equivalent Australian government regulations will be used for the sake of simplicity.

5 Literature Review

The literature review first approaches the matter of ontology learning, revealing several possible methodologies. Common to all approaches is the desire to reuse existing ontologies. As such, a survey of current IoT, Privacy and IoT-Privacy ontologies identifies the salient gaps and opportunities for novel contribution to the field. It concludes with a discussion on the IoT and Privacy domain's established research methodologies, highlighting current deficiencies.

5.1 Ontology Learning

Ontologies can be generated and maintained automatically (aspirationally) and semi-automatically (currently) through a process known as *ontology learning*. Ontology learning is a discipline that aims to apply knowledge discovery techniques to multiple data sources to support the task of developing and maintaining ontologies using a bottom-up, data-oriented approach[23].

Many frameworks for ontology learning exist. Arguably the most influential is the *text2onto* process outlined by Alexander Maedche and his colleagues in the early 2000s. Their ontology learning process occurs in four phases *Import and reuse, Extract, Prune* and *Refine*. The development of ontology learning approaches has continued since Maedche wrote the book on it in 2002. In 2007, Bedini and Nguyen proposed a different ontology learning cycle [28]: *Extraction, Analysis, Generation, Validation* and *Evolution*. They also highlighted that there are several possible approaches to ontology learning:

1. **conversion based** where an existing structured data source is effectively re-mapped to the ontological structure.
2. **Mining based** where data mining techniques (usually NLP) extract the information required for ontology generation.
3. **External Knowledge based**. Where an external data source outside the immediate corpus is referenced to enrich its sources and understanding. Wordnet, DBPedia and WikiData are commonly used external sources.

For example, the approach used to develop *OntoHarvester* uses wordnet and DBPedia to identify hypernyms and hyponyms to assist in concept and taxonomy generation [29].

4. **Frameworks** are the packaged toolsets that can be used to conduct ontology learning that incorporate multiple of the above elements.

One of the key priorities identified by Bedini and Nguyen was an urgent requirement to mine XML and other semi-structured data sources effectively. While ontology extraction and conversion are reasonably well understood, the Mining of non-semantic structures like XML files, network traffic captures, log files is not well understood. Some work has occurred into XML mining [30], but there does not appear to be a compelling example within the ontology learning field. The most likely source of techniques will come from the network analytics and cybersecurity field, where network forensics and traffic analysis at scale has been an ongoing point of research. The same principle can be applied to IoT environments like smart cities for cybersecurity purposes [31], so it will be a likely source of information to further this data source for ontology learning. Being able to effectively extract concepts from log files, network traffic captures, and schema documents will substantially support the ability to automate conducting ontology learning on the rich data sources offered by the IoT.

There are many techniques available to support the extraction, pruning and refinement processes of ontology learning. The use of external data sources like wordnet and DBPedia has already been discussed, but other techniques include Bayesian inferencing [32], NLP[33], [34], part-of-speech tagging [35] and deep learning[36]. The use of bootstrapping to enhance automated ontology generation has been established as effective [35] and raises important questions about which ontologies from within the IoT and Privacy domains would be suitable for bootstrapping future approaches. A recent survey of the field identified that though significant progress is being made within the field of ontology learning, it is not yet at a point of maturity, and particular effort should continue to minimise the requirement for human involvement and improve the precision of the concept extraction mechanisms.

Ontology learning offers a path to push the development of IoT and Privacy ontologies towards a more dynamic, autonomous process. Achieving a reliable autonomous ontology learning process will assist in maintaining the currency of ontological structures and within the specific domain of privacy impact assessments over IoT networks, account for dynamic, heterogeneous systems with complicated, varied legal and political frameworks imposed on them. The immediate challenge is determining whether or not a viable ontology learning approach can be applied to IoT data files. The following work to be done on how to determine what the optimal approach is to understand the questions about privacy that need to be answered by the ontology. A symbiotic relationship between the two, leveraging a scenario or competency question-driven approach

[37], [38] to ontology generation for the privacy aspects and a data-driven approach to IoT aspects could see a substantial improvement towards ontology learning from semi-structured data sources and advances made towards reducing the involvement of humans in the decision-making loop.

Both Maedche[23] and Bedini's[28] ontology learning approaches begin with the extraction of usable knowledge. This bootstrapping process continues to be a significant enabler to effective ontology learning[35]. Resultantly, understanding the current state-of-the-art in IoT ontologies and Privacy ontologies is an essential first step to answering the research questions:

5.2 Internet of Things Ontologies

A number of ontologies have been developed in an attempt to address the requirements of the IoT domain. This section will explore the key contributions, identifying key limitations and opportunities.

5.2.1 Foundation IoT Ontologies

One of the earliest attempts was the Semantic Sensor Network (SSN), developed in 2012. The SSN describes sensors, the act of sensing, and sensors' measurements, using a *Stimulus, Sensor, Observation* pattern [39]. Developed ***manually*** and published by the World Wide Web Consortium (W3C), it is widely used and is effectively the dominant mid-level ontology for the IoT / remote sensing domain. Two other ontologies sit at a similar level of granularity but are less popular. They are the *oneM2M Base ontology* [40], the *Smart Appliance Reference Ontology (SAREF)*[41] and the *W3C Thing Description ontology* [42].

The oneM2M base ontology is effectively a competing ontology to SSN developed by the oneM2M group, mainly parallel to the W3C SSN ontology. It focuses on establishing a set of standards that IoT designers can employ when developing new devices and applications [40]. Its first version was also published in 2012 and was developed ***manually*** by a working group of domain experts. It is used and extended much less frequently than SSN is in the recent literature.

The SAREF ontology was released in 2015 due to a project that formed a working group of a small number of domain experts to ***manually*** develop an IoT ontology using a bottom-up approach. The group started with the ontologies specific to known devices and sensors, and where unavailable, made assessments of the semantic coverage requirements to enable interoperability. Based on these ontologies, they extracted the common concepts and relations to build SAREF. They concluded their study by mapping SAREF to oneM2M and determining that one M2M had better coverage and that the be used their

ontology should extend to reflect oneM2Ms functionality[41].

The W3C *Thing Description Ontology* is a newer ontology developed ***manually*** in 2019. The Thing Description ontology is focused on describing the *semantic Web-of-Things* [42]. The *Web-of-Things* model is an evolution of the Internet-of-Things concept that imposes additional technical requirements to allow the 'thing' to integrate into semantic web services, for example, the use of RESTful APIs and JSON payload exchange to communicate over HTTP [43]. In more general terms, it is a deliberate attempt to convert the IoT heterogeneity into a WoT homogeneity by forcing data exchange to the HTTP(S) protocols [4]. The main limitation of the Web-of-Things model is that it requires a-priori knowledge of the IoT resources it will be using. Antonazzi and Viola posit that largely ontologies and semi-formatted data for WoT applications will be static, and this is largely incompatible with the dynamism that we know exists in the IoT [44]. The *Thing Decription* ontology takes a semi-complementary approach to the SSN ontology, where the SSN focuses on observations made by sensors. The Thing Description focuses on the 'things' themselves and how they connect. The difference is subtle but important. The mappings between the two ontologies are highlighted in the Thing Description ontology definition [42]. Of importance, given that TD is a relatively new ontological approach, there is substantially less work using it in the literature.

A comparison of *SSN, oneM2M* and *SAREF* (among others) was conducted in 2016. The findings of that comparison highlight that a major weakness of SSN was the lack of support for actuation activities. SAREF does not account for communicating the actual observations. Rather just the devices themselves and the earlier forms of oneM2M focused almost exclusively on the devices and services, at the expense of the observations [45]. The latest oneM2M release appears to now account for the communication of observations and the device and service data [40]. Several other gaps have been noted over time, though, and substantial research effort applied to extend these ontologies (particularly SSN) to meet the specific domain requirements. Highlights of those efforts are below.

5.2.2 IoT Domain Ontologies

The 2012 efforts to develop an *IoT Description ontology* focused on extending SSN to include support for concepts of actuation, IoT gateways and servers. Further, it added support for modelling IoT Services, Quality of Service (QoS), Quality of Information (QoI) and physical location. While they do not explicitly specify their ontology development methodology, this was likely done through a ***manual*** process, utilising the expertise of domain experts. A point of novelty is the desire to automate testing by introducing a test-ontology with pre-developed use-cases to reuse and standardise testing the efficacy [1].

One of the following significant developments was support for streaming data. The *Stream Annotation Ontology* (SAO) was developed in 2014 to address performance issues for annotating data streams on resource-constrained platforms with high-volume throughput [46]. It extended the SSN ontology and also added support for QoI and QoS representation. Their specific methodology for developing this ontology is also not disclosed. However, based on their early goal identification and the subordinate nature of the ontology in their overarching work to improve stream annotation performance is was likely a ***manually*** engineered, top-down developed ontology.

In 2015 the *OpenIoT* ontology was proposed to extend SSN to support cloud, and mobile environments, including the relevant QoS data for mobile connections [47]. They do not specify their development methodology. However, given the focus on adding a few niche features to SSN, a ***manual*** top-down approach was likely employed utilising domain experts. The code for this ontology is still available online[1] however has not been updated since 2015 at the time of writing. The acknowledgement that the IoT will not just consist of static devices is a significant development from this method.

2015 also produced the *IoT-Lite* ontology. IoT-Lite extends both the SSN and SAO ontologies, aiming to produce a more performant ontological structure that will enable real-time object discovery and near-real-time querying[48]. They define their own (***manual***) process to build dynamic, scalable ontologies. Of note, they contribute a method of parameter estimation using Mathematic Markup Language (MathML) that reduces the complexity of storing approximation formulae in triples. The approach was, in their testing, more performant than previous implementations. Their guidelines, though manual, offer a possible workflow that can be explored further for automation prospects. The critical deficiency in IoT-Lite is a lack of capacity to handle streaming data [49], which is anticipated to be a growing requirement as 5G permeates the IoT Domain.

IoT-O in 2016 revisited IoT ontologies from first principles using the *NeOn* ontology development methodology [50], which focuses on scenario-driven ontology development, applying usecases to derive the required semantics. It is a ***manual*** approach that relies on domain expertise to develop both the scenario and the ontology concepts and relations. IoT-O applies the modular approach advocated by Gruber in his work on ontology bridging to minimise ontological commitment [5]. It does this by identifying and extending the valuable elements of SSN, SAREF and oneM2M ontologies, and some others for location, power consumption service and lifecycle modelling. It extends novel work through the creation of the Semantic Actuator Network (SAN) model, which uses an *action, actuator, effect* pattern to complement the *stimulus, sensor, observation* model of SSN. The case for rebuilding IoT-O as a new IoT core ontology was to make it more lightweight, following the development of several very expressive, very slow ontologies over the preceding years. The modular approach for combin-

[1] https://github.com/ OpenIotOrg/openiot/

ing discrete but related concepts is of particular note and worth investigating further as a design option for this project.

2017 saw the first ontology to recognise the value of context-awareness for IoT explicitly. That is, being able to search for a device by physical or logical location, as well as just services provided. While previous approaches have included a location in their ontology [1], the approach here is more deliberate in realising the implications of that modelling. Further, IoT-Context extends SSN for modelling the sensors by adding in statefulness concepts to indicate whether it is connected, powered on etc. Further, it uses the GEO ontology to make location queryable [51]. The critical limitation of the IoT-context approach is that it does not account for the extensions of SSN that introduce things like support for streaming data.

In 2019, the SOSA ontology was developed as a lightweight implementation of SSN that replaces the *Stimulus, Sensor, Observation* core with *Sensor, Observation, Sample, Actuator* (SOSA). The new approach is event-centric and facilitates the modelling of mobile as well as static devices. It adds actuation and sampling beyond the capability of SSN, and moving towards a semantic web-of-things construct supports the use of web exchange formats like *JavaScript Object Notation - Link Data*(JSON-LD). They describe their development approach as being use-case driven, implying the use of the *NeOn* ontology development method (or similar) employed as part of the W3C Open Geospatial Consortium on Spatial Data in the Web [52]. The *NeOn* approach is, of course, ***manual***. The SOSA ontology offers significant promise as a start point for further development, having captured and implemented most of the lessons learned in the IoT ontology development space in the preceding decade.

The most recent contributions to IoT ontologies all implement or refer to SOSA. Cabarello et al. note that SOSA is suitable for representing sensor/actuator behaviour and that their solution is compatible with SOSA. However, they have pursued a very novel path of modelling IoT networks as Finite-State-Machines (FSM) using ontologies derived from oneM2M Base ontology, and the W3C Thing Description [4] that make it functionally quite distinct from SSN. The innovative use of FSM to recognise and track state offers some opportunities to explore further, particularly as they pertain to reducing complexity and tracking context of devices. Further work on the WoT approach more deliberately extends SOSA but notes a focus on modelling the physical and data-link aspects of an IoT network. Therefore, it may be cumbersome at the level of abstraction associated with the WoT [44]. Antoanzzi and Viola also note that while the modelling of IoT and WoT has come a long way, there remains a substantive gap in understanding the security and privacy implications of a prolific, connected network.

Finally, the recent *IoT-Stream ontology* extends SOSA to handle streaming data just as SAO extended SSN prior. It takes a more lightweight approach

to make the labelling of data more efficient [53]. To develop the ontology, they used the Stanford Ontology101 methodology [8] which is ***manual*** and top-down driven to develop their ontology.

Of the substantial work conducted in IoT ontology design in the last decade, the initial focus was developing an expressive ontology that could handle all of the emerging domain elements as they become relevant. That initial focus bore out SSN, SAO for stream data annotation, Context-IOT for context modelling, OpenIoT for cloud and mobile device modelling, IoT-O and SAN to cover actuation. As the breadth of ontological commitment grew, even modular bridging approaches became cumbersome. Hence, efforts increased to make more lightweight ontologies that only covered the essential elements, leading to IoT-Lite, SASO and IoT Stream. The Semantic Web of Things advocates, rather than pushing to reduce the number of concepts, advocate for abstraction, focusing on modelling the services and applications rather than the underlying network, sensors and actuators that control it. No matter the focus, or the level of abstraction, of the approaches that disclosed their development methodology, all of them applied a ***manual*** approach heavily reliant on the availability of domain experts to design and produce the ontology. Maintenance remains an ongoing requirement, yet few of the git repositories or ontology definitions have been updated more recently than a few years ago. There is an enormous opportunity to explore automation of the IoT ontology generation process, using bottom-up approaches that leverage the availability of data and the power of ontology learning techniques. A second identified gap consistent across these ontologies is a lack of awareness and concern for privacy and security. Some hint at it, but as a rule, the ontologies themselves cannot answer questions about security and privacy. That is ok. Reflecting on the principle of minimal ontological commitment [7], the addition of privacy considerations should be part of a modular approach, bridging ontologies together to allow the appropriate questions to be answered. However, to understand what degree of granularity needs to be modelled, we need to understand the basic privacy questions that need to be answered.

5.3 Privacy Ontologies

5.3.1 Pure Privacy domain ontologies

A broad background of the approach to defining privacy was outlined in the introduction. The culminating point of the philosophical work into privacy is that there is no simple definition, so a taxonomic, typologic, or other framework-based approach is most effective. Frameworks, taxonomies, and typologies are well suited to be represented semantically as an ontological structure.

There have been attempts to encode privacy principles. Several privacy on-

tologies try and address gaps, mainly from a cross-domain perspective, where privacy has been an incidental concern or resulted from exploring its impact on a particular adjacent domain. One of the privacy domain approaches is Gharib's novel privacy ontology [54]. The novel privacy ontology uses a bottom-up approach, conducting a systematic literature review and then using a ***semi-automated*** text mining approach to extract key concepts and develop an ontology meta-model in the Unified Modelling Language (UML). The resulting ontology could not be found. The model itself, being derived from academic papers that reference *privacy, ontology, taxonomy* or *privacy requirements* is essentially a synthesis of other work undertaken to generate ontologies and an extrapolation of common thought more than a deep requirements analysis. It is helpful because it shows that ***semi-automated*** methods like text mining can be applied to the privacy and ontology domains.

The legal interpretations of privacy have driven a number of the more recent privacy ontologies. PrOnto was driven by GDPR requirements and used a Design Philosophy approach manifesting in the MeLOn (Method for developing Legal Ontologies) methodology [14]. MeLOn is a ***manual*** approach that uses domain experts to develop an ontology to answer specific requirements. The PrOnto ontology appears to be limited in assessing privacy impacts in general terms due to its heavy grounding in legal reasoning. The same author led a W3C working group to ***manually*** produce the Data Privacy Vocabulary (DPV) [15]. The DPV highlights that the focus is still on achieving compliance with a particular ruleset rather than a more general approach to understanding privacy principles and harms. The lack of a general structure means that existing ontologies like PrOnto will not map well between jurisdictions and will struggle to apply more to something like the IoT, which is cross-border, multi-jurisdictional and unconstrained in many ways by geography.

The earlier approaches to understanding privacy in the context of IoT were from the cybersecurity community. Issues that impact both security and privacy formed a nexus through which to examine privacy. As a contemporary example of one of those issues, implementing access control, [55] requires consideration of confidentiality, authorisation etc., which manifest both in the security and privacy space. The survey by Qiu et al. highlights that there is no general set of privacy principles or harms that are in accepted use. There are competing documents, largely due to the geographic and political variation of the privacy requirements even among standards. Ontological approaches to cybersecurity in IoT, such as Mozzaquattro et al. [56], demonstrate that the cybersecurity domain derives their understanding in terms of *information assurance*. While information assurance is valid for IT security, it does not capture some of the nuances of privacy as examined by Finn [13], Solove[11], or Kasper[12]. It presents a gap for broader privacy impact assessments to inform policymaking or even risk management decision making.

There is evidence supporting the broad understanding of the requirements for

IoT privacy, as evidenced through the PISCES project from 2016 [16]. PISCES is a ***manually*** developed *privacy by design* framework aiming to reach a point where privacy policies to be dynamically generated and presented to users to inform them about what risks they are exposed to. The paper highlights some of the requirements for a privacy-preservation system and importantly outlines the *privacy by design principles* that will be useful to inform future work. One of the apparent limitations is that it is grounded heavily in GDPR compliance, meaning that its implementation will be anchored to those specific requirements and not a more generally applicable privacy framework for IoT.

5.3.2 Privacy-IoT Domain ontologies

Two key ontologies specifically address privacy in IoT. Several others exist, but these two are the most contemporary and relevant to the previous discussion of IoT ontologies. They are *IoT-Priv*[57] and *LIOPY*[58].

IoT-Priv extends SSN through IoT-Lite and is developed using a ***manual***, top-down, domain-expert driven *Ontology101* [8] methodology. It lists a detailed table of requirements approaching the clarity of competency questions. While articulating the principles is good, they are derived from a Canadian accounting privacy code of practice and perhaps lack universal application. The ontology is currently limited in expressiveness due to a perceived lack of privacy corpus material to derive a meaningful understanding of the concepts in the domain. The requirement for a lightweight ontology is a clear requirement that should be accounted for in future work.

The *Legal IoT Privacy Ontology* (LIOPY) was developed in 2018 using an undisclosed methodology. However, it is likely to be a ***manual***, top-down domain-expert driven approach. The ontology itself extends SOSA and, as a result, is likely to handle streaming data poorly, having not yet inculcated the IoT-Stream elements. One of the significant contributions of the LIOPY is recognising that there is a data lifecycle. Data exists in use, in transit and at rest and must be accounted for in each phase from a privacy perspective. The ontology uses GDPR compliance as its basis for use and applies an approach of identifying and applying privacy policies using inferencing driven by description logic. LIOPY provides a strong starting point for future work on privacy ontologies.

Privacy is a slippery concept without a convenient definition. It is best expressed through a typology or taxonomy or another nuanced framework from a philosophical standpoint. The current work towards developing a shared understanding of privacy vocabulary and requirements is hindered by variations in privacy requirements between areas of geographic and political influence. As a result, being the most notorious data privacy regulation framework, GDPR compliance is a driving force behind most literature. The desire to meet those

specific requirements has shaped the development of both legal and IoT ontologies towards a compliance framework. There is an opportunity to explore privacy implications for IoT through a broader lens of privacy impact assessment informed by the philosophical models of privacy. The nuance here may better enable policymakers and risk assessors to decide how an IoT network is likely to manifest privacy concerns across the whole data lifecycle.

5.4 Gaps (potential research contributions)

1. **Automated or Semi-Automated IoT Ontology Generation**. Throughout the vast majority of the ontologies examined in this literature review, a top-down, domain expert-driven approach to ***manual*** ontology creation has permeated, if the methodology was specified at all. While some of those approaches are methodical and standards-based like *Methontology*[18], and *Ontology101*[8]. Some of them are more focused on developing ontologies to solve specific use-case problems, like NeOn [50]. Regardless of approach, they are all ***manual*** and develop static ontologies that require domain experts to update them manually as the domain that they represent changes. That is, *Ontology learning approaches have not yet been applied to the IoT Domain*

2. **Balancing expressive and lightweight ontological structures**. The ontology learning processes all rely on a deliberate pruning phase after the initial extraction/creation phase. The manual approaches advocate for beginning with minimal commitment and utilise *competency questions* as a mechanism to define the requirements for an ontology to be *done*. There is no evidence in the literature of anyone attempting to incorporate automated competency question evaluation as a primary pruning mechanism to improve the efficiency of the ontology learning process, balancing the development of an expressive ontology with a minimal ontological commitment that characterises lightweight implementations.

3. **Ontology learning from semi-structured data**. There have been no meaningful attempts to conduct ontology learning tasks utilising semi-structured data such as log files, network traffic data packet captures, or an extensible markup language as an input. *Any work in this area will achieve novelty*

4. **Abstraction of IoT Privacy from specific legal frameworks**. They are investigating the *privacy* domain also that the privacy requirements will change between legal jurisdictions. Though, the existing ontologies are all founded in a specific legal or policy framework (mostly GDPR). That is, *ontology learning presents an opportunity to automate the generation of a privacy domain ontology that is either general enough to apply everywhere or is flexible enough to be tailored to each domain*

6 Methodology

The section on research methodology will begin with examining the philosophical underpinnings of the selected research approach, highlighting how it addresses the specific research problem. Based on the determination of the approach, the specific methodology chosen will be explained.

6.1 Philosophical foundations of research methodology

Selecting a research approach depends on the nature of the research problem and the intended audience who will review it [59]. This research problem is rooted in domains that have been discussed here extensively as being characterised by *complexity uncertainty* and *dynamic change.* Further, the ties to knowledge and ontological engineering highlight that some kind of *modelling* (here, meaning to make an artificial representation of the real world) needs to occur. The approach needs to be effective at dealing with the inherent *complexity uncertainty* and *dynamism.* Previous research into Information Systems sensemaking has suggested the use of the *cynefin* framework to handle *wicked problems.* A *wicked problem* is one that is *ill defined, dynamic* and *complex* enough that it required a holistic approach to solve it [60].

The Cynefin framework articulates five contexts. The two unordered contexts are *Chaos* where we are unable to make sense of what is occurring, *complex* where cause and effect are unclear but can be derived in retrospect. The two ordered contexts are the *complicated* where we sufficiently understand the world to build things and *simple* where we understand and can fully predict cause and effect [61]. The fifth is *disorder* and exists when it is unclear which of the four domains currently applies. Interestingly, each of the four domains necessitates a different approach. The *Chaotic* domain requires us to act (often randomly), observe the result of actions and then respond to try and make sense of what is occurring. *Complex* systems require an initial probe, a deliberate action based on a hypothesis that we then sense the responsive action to address. *Complicated* systems are understood well enough that we can observe them, analyse what we see and then respond. Finally, *simple* systems, we can observe and categorise the behaviour we see into what we already understand [62]. In general terms, the goal of a decision-maker interacting with a system through the lens of the cynefin framework is that they move the observed system from being chaotic to complex, to complicated to simple. Simple systems resolve to business rules and processes that require little thought.

In mapping this research problem to the cynefin framework, it is immediately apparent that we are not dealing with a *chaotic* system. Some rules exist, and we have an understanding of the involved entities. However, the system is also not simple enough to be *business as usual* in the *simple* domain. The

key difference between the *complex* and the *complicated* domains is that in a *complicated* domain, cause and effect are evident with observation and analysis. In a complex domain, it is only evident after the fact. Currently, the IoT and privacy domains independently are probably resident in the *complicated* domain. However, the interaction between those two domains pushes quickly into the *complex*. We are unsure how the IoT devices impact users' privacy but could conceivably start from a breach and trace it back to its origin. This research project will identify a probe (research question), assess its results and then adapt our understanding of the domain accordingly. Of particular interest is that the majority of modelling approaches (inclusive of manual ontological engineering approaches favoured heavily in the IoT Domain) are only effective in the *complicated* and *simple* domains. In order to model something, we need to be able to understand the cause and effect associated with it. Techniques like exploratory data analysis, multivariate statistics and *data mining* however are suited for use in the *complex* domain, as they have no requirement for *a-priori* knowledge [63]. The ontology learning techniques identified by Maedche heavily utilise data mining, and hence, are more suited to use in the IoT-Privacy problem than the manual approaches currently favoured.

When requirements are unclear in the software development arena, there are typically many unknown factors, and the situation is dynamic. It best practice to adopt an *agile*[2] development approach. Agile approaches combat the *complex* environment. They scope small time boxes, develop and deploy an artifact (probe the system), get user feedback (observe the response), and refine their approach based on their feedback. When we consider the *wicked problem* descriptor, this approach addresses the attributes of those problems and is reflective of the approach suggested to progress through the *complex* domain. While several deliberate research methods can apply an agile approach to sensemaking, the one selected here is the *design science research methodology* (DSRM), noted for its effectiveness in addressing complexity [64].

6.2 Practical implementation of research methodology

DSRM is a research methodology that focuses on the holistic and systematic approach to the design and development of an artifact to fill a research gap [65]. In more general terms, it is a research approach grounded in pragmatism that may be characterised as a form of *applied* research. One layer of specificity down, it can be characterised as part of the computer science *build* research methodology [66]. A research project employing a *build* methodology (such as DSRM) consists of building an artifact, either a physical artifact or a software system, to demonstrate that it is possible. To be considered research, the construction of the artifact must be new, or it must include new features that have not been demonstrated before in other artifacts[66]. The build methodology, as well as

[2]https://agilemanifesto.org/principles.html

most implementations of DSRM follow a variation of the following principles: *design an artifact; Address a relevant problem; evaluate the design; make a clear contribution; apply rigour; the design is a search process;* and *results must be communicated* [64]–[67].

Offerman's Design Science Research Methodology will be used as the framework for this research [65] and is shown in Figure 1.

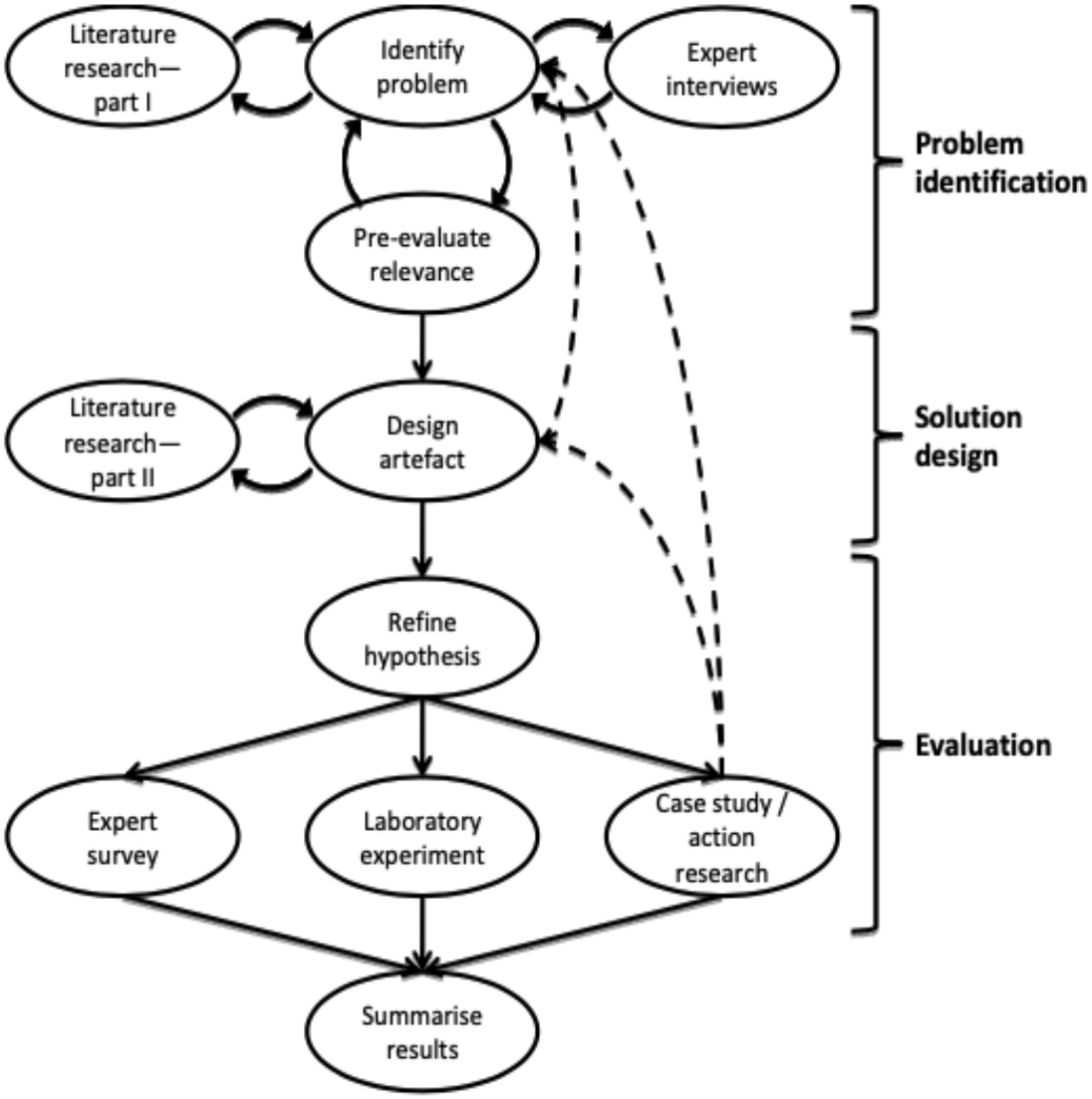

Figure 1: Design Science Research Methodology

Summary. The research problem examines how to build an ontological structure for the complex intersection of two complicated domains. Specifically, it is looking at how to employ *ontology learning* to solve those problems. Based on our assessment of the domain complexity, the manual ontology development methodologies heavily relied on in the IoT domain will not be suitable for examining the intersection between the two domains. Instead, an agile approach that can probe, sense and respond is needed to address and reduce the complexity. A design science research methodology is the mechanism best suited to guide this research project. While the detailed implementation plan will be explained in the next section, the approach essentially involves identifying a problem, building a solution, and evaluating rigorously to determine if the developed solution can address the research gaps.

7 Plan

Offermann's DSRM approach consists of three broad phases, each of which has a number of activities:

1. **Problem identification.** Needs to confirm that the problem is relevant and solvable
2. (a) **Identify problem**. A specific, but substantial problem must be identified.
   (b) **Literature Research I**. Review the state of the art and confirm the gap.
   (c) **Expert Interviews**. An optional step to assist with the verification of the problem.
   (d) **Pre-evaluation relevance**. Confirm understanding of the problem and generate a research hypothesis.
3. **Solution Design** Builds a candidate solution to the identified problem
4. (a) **Design artifact**. A creative engineering process to design and build s solution to the identified problem
   (b) **Literature research II**. Ongoing study as the specific detail of the design and build unfolds.
5. **Evaluation** Demonstrate the applicability of the solution to the identified problem.
6. (a) **Refine hypothesis.** Break the general hypothesis (from the *pre-evaluate relevance* step into smaller hypotheses which together support achieving the primary hypothesis.

(b) **Case Study / Action research**. One of the three mechanisms to evaluate the solution. Involves applying the solution to a real problem or case study and evaluating its ability to solve that problem against the hypotheses set.

(c) **Laboratory Experiment**. The second of the three evaluation mechanisms. Involves deliberate experiments to compare specific features of the developed solution to existing solutions to compare functionality.

(d) **Expert Surveys**. The last of the three evaluation mechanisms. Involves exposing the solution to experts and asking for their assessment of it

(e) **Summarise results**. The most important component of the process, collating and communicating the findings.

7.1 Agilitology Approach

Given the above discussion about developing a candidate solution to *automate privacy impact assessment for the IoT* against the backdrop of complexity identified in section 6.1, an approach that combined formal methods and an agile approach was required. Previous work on cyber-security domain ontologies has employed the *Agilitology* approach [22]. A detailed definition of the Agilitology approach is available [68], but the core of it relevant to this work is that it sits at the intersection of the *formal, manual* ontology development approach *methontology*[69], DSRM[67], and TDD[70] and produces an ontology that is heavily biased towards a *lightweight* implementation. It achieves this by only adding a concept, property or relation to the ontology where there is a driving *competency question* that requires the inclusion of the concept, relation or property to answer the question. The Agilitology approach inverts traditional manual ontology development approaches which tend to try and formally capture all domain knowledge up-front and as a result incur significant development time and costs, and require access to both deep domain knowledge and ontology engineering skill.

Generating an ontology by starting with the domain use-case holds some risk of *over-fitting* to the use case and generating a very specific *application ontology*, as described by Maedche[23]. In ontology-engineering terms, this is an ontology which displays a *heavy encoding bias* and a *heavy ontological commitment* [7]. While there is nothing inherently wrong a specific application ontology, developed using a specific encoding scheme (here: RDF Turtle), where we are intending to generalise the result (i.e. use this ontology to represent more than just a single dataset) it can be problematic. This is where the superimposition of the *TDD* and *DSRM* approaches over the *Methontology* approach in *Agilitology* is useful. The constant revision and refactoring of the solution pushes the developer to start with a problem (the Competency Question), make it work

(add the concepts, properties and relations to make it work) and then make it work good (prune and refactor the concepts, properties and relations to be more generalised). Or, in Beck's *TDD* language *"Red, Green, Refactor"* [70].

The resulting ontology that is built using the *Agilitology* approach is one that abides by Gruber's principles of ontology design (*Clarity, Coherence, Extensible, Minimal Encoding Bias and Minimum Ontological Commitment*[7]. Importantly, it is developed quickly. Given the underlying business problem described in 4.1 being that there is currently no mechanism to assess privacy impact on 5G IoT networks, development of a minimally viable product to enable further research and experimentation to continue is a driving imperative. The Agilitology approach produces just that, a minimum viable product, or a *minimally viable (lightweight) ontology* if you prefer.

7.2 Summary of Research Methodology

The DSRM process has guided the completion of the research project documented in this thesis. Throughout the design and refinement of the artefact, the *Agilitology* ontology development method was adopted to combine the benefits of a deliberate methodology (*Methontology* with the features of *TDD* and *DSRM* that produce a *minimally viable ontology* that forms a lightweight core for future work.

The resulting ontology and knowledgebase developed using the *Agilitology* method are described in the next section.

8 The Privacy Impact Assessment Nexus Ontology (PIANO)

This chapter is broken up five components. First, is an overview of the *Privacy Impact Assessment Nexus Ontology* (PIANO) with a short introduction to the use-case driving the development of the ontology and associated knowledgebase. Next is an overview of the *privacy* namespace, then the *sense* namespace followed by a brief discussion on *ontology bridging* in PIANO and finally an explanation of how the *Social Internet of Things Knowledgebase* was developed by applying the PIANO to the *Social Internet of Things* dataset.

8.1 Overview of PIANO

This section proposes an ontological structure to partially answer the question *"What is a suitable ontology development method to be used to derive the semantic links between IoT device data and privacy concepts."* Analysis in sections 5.3.2 and 7.1 have highlighted a driving requirement to use automation where possible to reduce the manual maintenance burden and to conduct any complementary manual development in an *agile* manner to focus the development towards a *minimally viable ontology* that can meet the needs of the system, with a bias towards a lightweight, extensible structure.

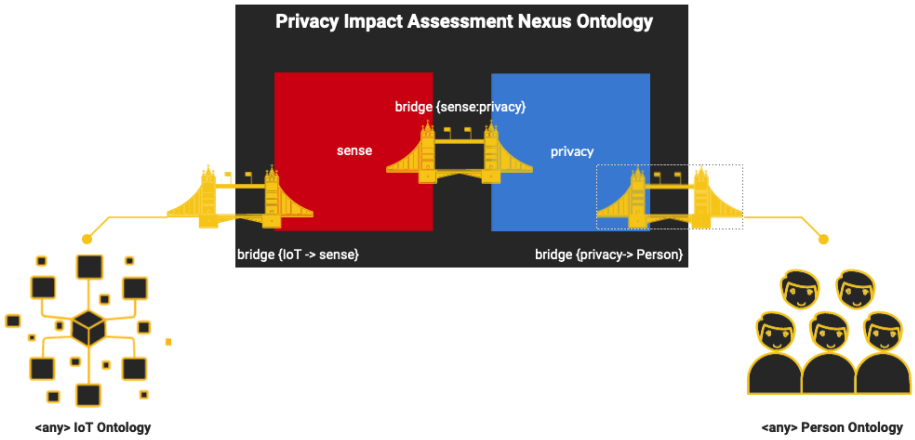

Figure 2: PIANO Context Diagram

Shown in figure 2 PIANO is comprised of two subordinate namespaces: ***privacy***, which focuses on the individual persona and privacy impacts and ***sense*** which focuses on the IoT Device and quantifying risks associated with IoT collection vectors. The PIANO also has three notable *ontology bridges*. An *ontology bridge* [71] is an ontological structure which is used to connect disparate ontologies. These will be covered in more detail below.

8.1.1 Purpose of PIANO

The purpose of PIANO is to serve as an abstraction layer that connects ontologies representing physical IoT networks to ontologies representing the people interacting with them, enhancing the abstracted connection with novel insights about the privacy impacts where those two domains interact.

8.1.2 Development Approach

The development of PIANO followed the *Agilitology* approach described in section 7.1. More detailed descriptions of the development approach are in the following sections and synthesised in the conclusion, but in broad terms the approach was:

1. Develop a series of competency questions about the *privacy* namespace based on the selected privacy framework.
2. Identify suitable dataset to serve as a use-case to drive the *action research* component of the research methodology to inform competency questions for the *sense* namespace.
3. Identify a number of competency questions that focus on the interaction of the two namespaces
4. Use *Agilitology* to build the initial *privacy* and *sense* namespace representations, derived from the competency questions.
5. Instantiate the dataset into a *knowledgebase* using the existing namespace schemas
6. Refactor the namespace schemas to accurately represent the usecase
7. Design and automate the execution of a number of queries to construct implicit relations within the knowledgebase based on the underlying namespace schemas
8. Conduct exploratory data analysis on the resulting knowledgebase, to evaluate the efficacy of the process
9. Refactor where required to achieve a generalised solution.

8.1.3 Implementation

The application of the *Agilitology* approach to ontology design means that there is an intrinsic link between the source dataset usecase and the implementation of the ontology itself. The iterative *TDD* and *DSRM* facets of the research methodology mean that for each of the namespaces, an initial estimate of functionality was conducted. This was achieved by the development of the following competency questions, which were then used to drive initial schema construction:

1. Privacy namespace, derived from the privacy ontology literature review in section 5.3, in particular from the work of Finn et. al.[13]:

(a) WHAT are the different types of privacy?

(b) WHAT are the different vulnerabilities someone has to be collected on?

(c) WHAT are the relationships between privacy types and a person's vulnerabilities?

2. Sense namespace, derived from the common features of privacy-aware IoT Ontologies identified in the literature review in 5.3.2:

   (a) WHAT are the different vectors for collection?

   (b) WHAT are the specific mechanisms a device can use to collect information?

   (c) WHO owns a given device?

   (d) WHAT vectors is a given device collecting?

3. Internal ontology bridge:

   (a) WHO is a given device collecting on?

   (b) HOW does a device compromise an individual's privacy?

   (c) HOW do different levels of compromise impact different types of privacy?

From these competency questions, an initial set of concepts, properties and relations were constructed. The remainder of the development steps will be described in subsequent sections, with the exception of the *ontology bridges*.

In PIANO, the *internal* ontology bridge is used to connect the *sense* namespace to the *privacy* namespace. That is, it connects the physical and virtual worlds containing the IoT Devices and their data to the conceptual 'privacy impact' world.

8.1.4 Summary of PIANO

The Privacy Impact Assessment Nexus Ontology (PIANO) forms an abstraction later that connects existing IoT Ontologies to existing People Ontologies, enriching that connection with novel contributions about the privacy impacts of their interaction. It was developed using the *Agilitology* ontology development method following the broad steps outlined in section 8.1.2. It provides a significant step towards solving the *business problem* identified in section 4.1, the scientific problem identified in section 4.2 and answering research questions 1, 2 and 3 listed in section 4.3.

8.1.5 Introduction to the Social Internet of Things Dataset (Usecase)

The *Agilitology* approach necessitates the use of a usecase to drive development. The initial conceptual usecase was provided by competency questions derived from the literature review, outlined in section 8.1.2.

After an initial review, the UNSW TON-IOT dataset[26] was not pursued further. This is largely because the dataset was primarily comprised of internal logs generated by IoT devices and lacked sufficient information to determine likely collection vectors. The TON-IOT Dataset will likely become useful in the future, when PIANO is extended to handle cyber security incidents, and their privacy impacts.

The *Social Internet of Things Dataset*[27] was selected as the dataset to generate the usecase from. The decision is based on the representation of distinct users, devices and indicative services providing a sufficient depth of information to generate a meaningful knowledgebase of interactions.

8.1.6 Overview of Social Internet of Things Dataset

The *Social Internet of Things Dataset*[27] is available on their website, and copies of the relevant files have also been stored on the Author's github for posterity (and version control of data).

The subset of the Social IoT Dataset used here consisted of the following three components:

1. **Social IoT Objects Description** which describes an IoT Device's associated user and the type of device (e.g. smartwatch, garbage truck).
2. **Social IoT Objects Profiles** which describes the types of 'services' available on each device.
3. **Service and Application Description** which describes the numerical encoding of the two above data sources

Overall the dataset provides

	Count	Comment
Users	4000	Mix of Public and Private
Devices	16216	14,000 Private, 1616 Public
Device Types	16	8 Private, 8 Public, Allocated by % to user
Services	7	Assigned to Device Type. Applications excluded.

The Dataset contains substantially more data points beyond these, but given the abstraction layer (away from the physical world) they have been excluded from the generation of the *Knowledgebase*

8.2 The *privacy* Namespace

The privacy namespace is the abstract representation of the concepts, properties and relations pertaining to the *people* and their *privacy*.

8.2.1 Purpose of the *privacy* Namespace

The privacy namespace sits between two ontology bridges. The internal bridge, to the *sense* namespace is where the focus of development is weighted, and is described in detail in section 8.4. The external bridge is from the *privacy* namespace abstractions of the people to the "physical", granular representations in another ontology.

In addition to providing this bridging function from the *sense* namespace to the physical representation of people in a more granular ontology, it also services to enrich knowledge traversing that bridge with information about the privacy risks associated with those 'people' and their interactions with IoT Devices.

8.2.2 Design of the *privacy* Namespace

The development process for the *privacy* namespace is detailed in section 8.1.3. The iterative improvement of the ontological structure following the guiding *Agilitology* approach has resulted in the ontology shown in figure 3 below:

The design of the namespace saw moderate change throughout the development process, with the "Persona" concept to represent an individual replacing an initial "Interrogative Dimensions" concept, with sub-classes of "Who, What, When, Where and Why". While iterating through the development of the ontology as part of the *Agilitology* process by adding the usecase to the ontology initially developed from the competency questions listed in section 8.1.3 it became apparent that there was a need for a concept to represent the target of the collection by an IoT device, and whose privacy could be compromised. The two concepts in the privacy namespace are described in further detail below.

The "Privacy Type" concept The *Privacy Type* concept were developed early in the *Agilitology* process, driven by the competency questions listed in section 8.1.3. They are derived from the Privacy Model outlined by *Finn et. al.*[13], which was chosen for two reasons. The first is that it focuses on

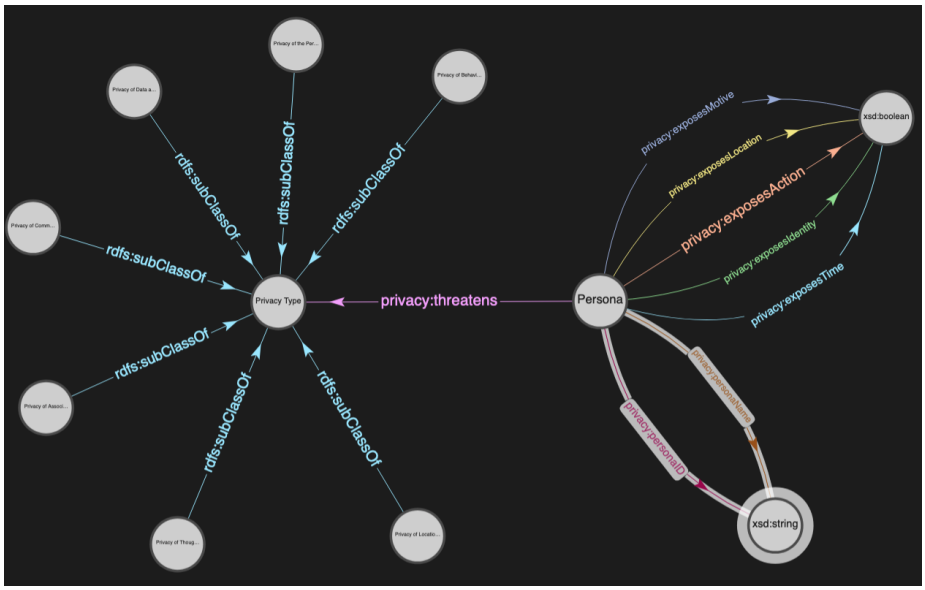

Figure 3: Privacy Namespace (Stardog Studio Capture)

the types of privacy, rather than the types of privacy compromise. The focus on privacy itself ensures that the ontology will be less *ontologically committed*, in line with Gruber's design principle and hence it will be less vulnerable to disruption by technological change. The second key reason it was chosen as the basis for the *Privacy Types* concept is Independence from any particular legal system. A heavy reliance on GDPR compliance is a major limiting factor for many existing privacy ontologies, and so an independent model removes the key barrier to international adoption and allows for an truer assessment of *privacy risk* lens rather than the *privacy compliance* lens that any framework tied to a specific legal system encourages. Detailed descriptions of each of the concepts is available in the original paper by Finn et. al.

The "Persona" concept The Persona concept was added to the privacy namespace midway through development. It represents the "person" entity within the scope of PIANO and has the associated properties of *personaID* and *personaName*, both string values used to uniquely identify a user, and list their human-readable 'name' respectively.

In addition to the string properties, a number of boolean properties belong to the Persona concept. Namely *exposesIdentity, exposesAction, exposesTime, exposesLocation* and *exposesMotive*. Each of these variables can be set to *true* where the individual conducts themselves in a way that presents a risk to each respective dimension of their persona. The dimensions are derived from the commonly used *interrogative dimensions*, namely Who (identity), What (ac-

tion), When (Time), Where (location) and Why (Motive). Considering a persona's vulnerability in these dimensions allows us to conduct in-depth analysis of where an individual can change their behaviour to reduce risk, what the most commonly displayed vulnerabilities across a sample are (e.g. within a company or geographic region) and sets conditions well for future work on recommender systems to prompt users for how to reduce their vulnerability to collection activities by IoT networks.

The overarching relationship between the *Persona* concept and the *PrivacyType* concept is the *threatens* relationship. While the detail of how these relationships are generated is described in section 8.6.4, the general idea is that a *threatens* relationship will only be created when certain criteria regarding combinations of *Persona* compromise are met, actualising the threat.

8.2.3 Implementation of the *privacy* Namespace

Detail of how the *privacy* namespace is implemented is described in section 8.6 where the creation of the *Social IoT Knowledgebase* is described.

8.3 The *sense* Namespace

The sense namespace is the abstract representation of IoT Devices, their ability to collect information as well as the properties and relationships associated with these concepts.

8.3.1 Purpose of the *sense* Namespace

The *sense* namespace, like the *privacy* namespace, sits between two ontology bridges. The internal bridge, to the *privacy* namespace is where the focus of development is weighted, and is described in detail in section 8.4. The external bridge is from the *sense* namespace abstractions of the IoT Devices to the "physical", granular representations in any of the IoT ontologies described in section 5.2 of the literature review.

8.3.2 Design of the *sense* Namespace

The development process for the *sense* namespace is detailed in section 8.1.3. The iterative improvement of the ontological structure following the guiding *Agilitology* approach has resulted in the ontology shown in figure 4 below:

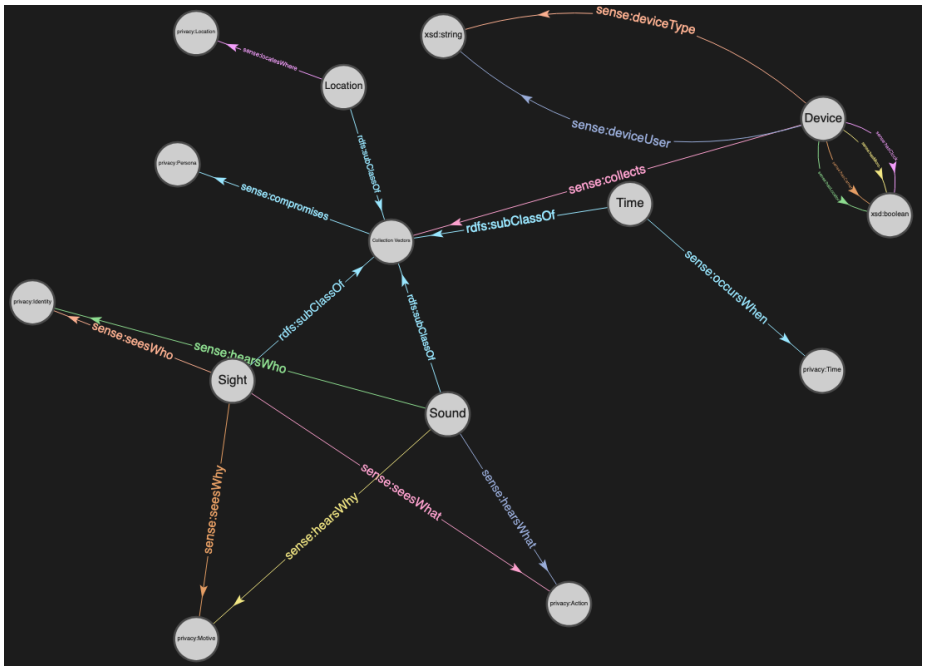

Figure 4: Sense Namespace (Stardog Studio Capture)

Unlike the design of the *privacy* namespace which was driven largely by theoretical privacy frameworks, captured in competency questions, The design of the namespace was driven heavily by the content of the Social IoT Dataset use-case. Initial analysis of the dataset highlighted a number of interesting questions that a privacy researcher would want to be answered that were no immediately apparent from the available data. These questions formed the bulk of both the *sense* namespace competency questions and the *ontology bridge* competency questions. To mitigate the risk of *heavy ontological commitment*, the *refactor* aspect of the TDD dimension of the *Agilitology* approach focused on abstracting as many of the concepts as possible, so that a minimal amount of detail would be required from the 'granular' physical IoT ontology to inform the analysis of PIANO.

The "Device" concept The *device* concept is the *ontology bridge* anchor that is intended to link from PIANO to granular, physical IoT ontologies. Its key identifying properties are *deviceUser* and *deviceType*, which are string variables respectively intended to link the device to an owner (here the link to the *Persona* concept) and identify the type of device (e.g. smartphone, printer, personal computer). Each device also has a number of boolean variables denoting the presence or absence of specific device features. Presently a device may also: *hasCamera, hasMicrophone, hasClock* and *hasLocator*. There are obvious omis-

sions of common device features here (e.g. logically, *hasAccelerometer* would be included, but given the lack of first-order privacy impacts from an accelerometer it was scoped out in this first *minimally viable ontology*. The intent is to demonstrate that common device sensors can be modelled abstractly in the ontology, and adding them into future iterations of the model is only a minor change to device properties in future.

The "CollectionVector" concept The *CollectionVector* concept is intended to serve as an intermediary point between the *Device* and the *privacy* namespace. Here, the different collection vectors *Sight, Sound, Location* and *time* are derived from the sensory dimensions *Sight* and *Sound*, and the key datapoints provided by the usecase data (*location* and *time*). When an IoT device is able to use *smell, taste* and *touch* to threaten privacy, there is scope for it to be extended past its current *minimally viable ontology* status.

The overarching relationship between the *Device* and *CollectionVector* concept is *collects*, indicating that the particular sensors on a device are capable of collecting certain information about the world around them.

8.3.3 Implementation of the *sense* Namespace

Detail of how the *sense* namespace is implemented is described in section 8.6 where the creation of the *Social IoT Knowledgebase* is described.

8.4 Ontology Bridging within PIANO

As described in section 8.1, an ontology bridge is a device that is used to connect disparate ontologies. Within PIANO, there is one internal bridge between the *sense* and *privacy* namespaces, and then two external bridges between the namespaces and their real-world equivalents.

The first external ontology bridge is in the *sense* namespace and is anchored on the *device* concept. The purpose of this ontology bridge is to move between 'abstraction' layers. Abiding by Gruber's *extensible* principle, the *device* concept is a (considerably simplified) abstraction of the physical IoT Device. Therefore, when connecting PIANO to any physical IoT ontology, the point of interface will be the *sense:device* concept. There are two key benefits to abstracting device here. The first is that an abstracted concept here reflects an effort to minimise *ontological commitment* and to *reuse existing ontologies*. As 5.2 highlighted, representations of physical IoT networks in ontological structures is well developed and there is limited value in contributing yet another slightly different representation of the physical world. The result of this choice is that is can effectively extend *any* existing IoT ontology with some small development work

to map the key properties required by PIANO from the physical IoT ontology. The second benefit is that abstraction here allows us to optimise query time. The less knowledge that is encoded into the ontology, the simpler queries are and the more efficient it is to use. Given the highlighted needs to favour lightweight approaches and edge-processing identified in 3.1 it is consistent efforts to solve the driving business need described in 4.1.

The second external ontology bridge is in the *privacy* namespace and is anchored on the *Persona* concept. Here *Persona* refers to an abstract representation of someone's 'persona', which is, effectively, a representation of how they are perceived in the world (whether physical or virtual). The bridge is intended to flow from the abstract *Persona* to an ontology that physically represents people, organisations and their interactions. Bridging here will enable interesting analysis to be conducted in those more 'physical' ontologies about how compromised privacy impacts the individual and their social networks. The benefits of this approach are in-line with those of the first ontology bridge.

8.5 Summary of the Privacy Impact Assessment Nexus Ontology (PIANO)

The Privacy Impact Assessment Nexus Ontology (PIANO) has been developed using the *Agilitology approach*. It is heavily driven by the *action research* principle of solving a real-world problem, here - how to assess the impact of IoT devices on personal privacy in the *Social IoT Dataset* usecase. The iterative approach to solving this problem has resulted in the development of a *lightweight*, *minimally viable ontology* that is specifically designed to bridge the current gap between the IoT and the people that use them. Better yet, it will augment that bridge with novel information about the privacy threats associated with that relationship and in doing so makes a significant step towards solving the business problem specified in section 4.1, the scientific problem articulated in section 4.2 and makes a genuine contribution to answering the research sub-questions b and c.

8.6 The Social Internet of Things Knowledgebase

The Social Internet of Things Knowledgebase is a novel contribution of this research. It uses the dataset from the *Social Internet of Things* project [27] and uses it to both drive the development of PIANO, and demonstrate the utility of the contribution in the *evaluation* of PIANO.

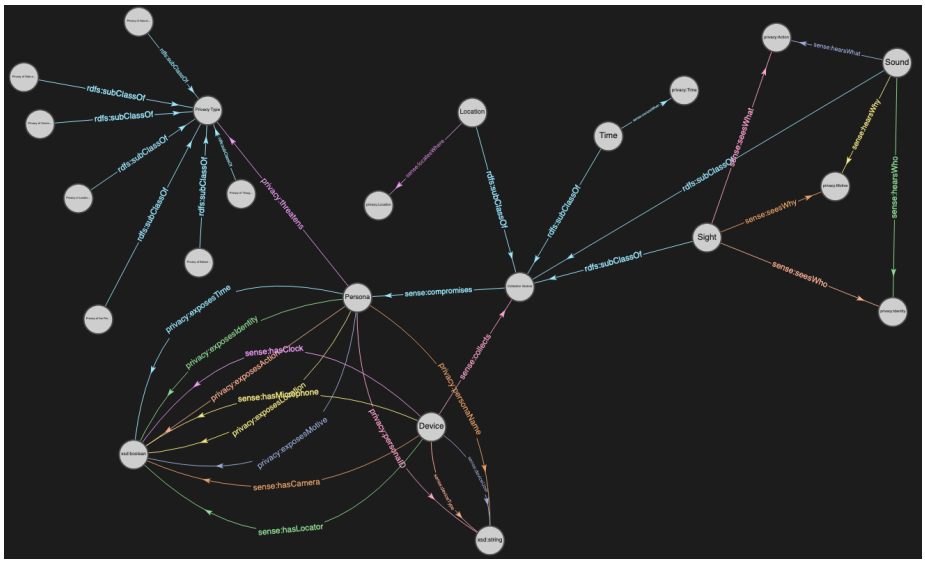

Figure 5: PIANO (Stardog Studio Capture)

8.6.1 Purpose of the Social Internet of Things Knowledgebase

The Social Internet of Things Knowledgebase is designed to hold the data from the Social Internet of Things Dataset when it has been instantiated using the PIANO schema. It is the location where novel insights about the privacy threats linking devices and personas are generated using *semi-automated ontology learning* and demonstrates the real utility of PIANO to addressing current research gaps in assessing the privacy impacts of Internet of Things devices.

8.6.2 The Social Internet of Things Knowledgebase Instantiation

The Social Internet of Things Knowledgebase is instantiated by a Python Script located on the Author's Github (along with relevant YAML file that can be used to construct an appropriate virtual environment using Anaconda.

The following functions (only invocations and descriptions shown here for space) take the input .csv files and transform them into a python dictionary of dictionaries, containing all the required information about the IoT Devices ans Users.

```
1 def ImportData():
2   #This function imports the raw data from the Social IoT Dataset
      and holds it in memory for processing by the "CreateDeviceDict"
       function.
3
```

```
def getRelevantItems(fullServiceList):
  #This function takes the list of device types & services from the
     object_profile input and trims the list to only the relevant
    values (1-9) of availiable services. "Apps" are deliberately
    discarded, and the device type is contained in the devicesDict
    dictionary the information is being passed into in "
    CreateDeviceDict" function.

def CreateDeviceDict():
  #This function takes in the raw data from the CSV and transforms
    it into a Dict that can be serialised into the privacy ontology
     graph.
```

From there, the Python script transforms the data in the devices dictionary data structure into RDF Triples, and pushes the triples to a locally running Stardog instance:

```
def createKBTriples(deviceDict):
  #This function is designed to turn the data held in the
    deviceDict dictionary into a series of RDF triples to be
    populated into a knowledgebase.
  #It accetps an input of a dictionary of dictionaries describing
    each device in the Social IoT.
  #It returns no value, and works as a frameword for other
    functions to achieve the data manipulation.

def makeRDFTriple(DeviceID, DeviceType, UserID, ServicesList):
  #This function is designed to create RDF triples out of various
    string and list values derived from the Social IoT dataset.
  #It accepts string inputs of DeviceID, DeviceTye & userID, and a
    list input of Services List.
  #It returns a formatted string that defines the properties of a
    SINGLE device from the Social IOT dataset.

ef makeTriplesForPersona(UserID, nameDict):
  #The purpose of this function is to create Persona entities to
    instantiate the "social_IOT_KB knoweldgebase"
  #It accepts an input of a UserID (Derived from the social IOT
    Dataset User ID) and generates a random name (acting as a
    placeholder here)
  #From there, it creates the RDF triples to create a persona
    entity and its related properties.
  #It outputs an RDF Triple as a formatted string.

def createTTLFile(deviceTriples, personaNames, firstTimeFlag):
  #The purpose of this function is to combine the device triples
    and the persona triples into a Turtle syntax
  #RDF file that is used to instantiate the Social IoT
    Knowledgebase.
  #It is called iteratively for each Device & user, and accepts a
    TTL formatted Device entity String,
  #A TTL Formatted Persona entity string and a boolen to determine
    if this is the first time through the loop
  #or not. If first time, it'll drop any old files & replace with
    the new.
  #The file returns no value, but does output the Device and
    Persona details to a TTL file on disk.
```

```
24
25 def createSocial_IOT_KB(connection_details, database_name):
26   #This function is designed to create the Social IOT Knowledgebase
        from available TTL Files.
27   #It accepts as an input the connection details and database name
28   #It assumes that the .ttl files are contianed in the same
       directory as the python script
29   #It returns no value, but does create & populate a database on
       the stardog server detailed in "connection_details"
```

Once these functions have executed, the Social IoT Knowledgebase exists in its proto-form, with the key entities and properties defined. During the execution of these functions, the data are enriched with several additional details, some from other parts of the *Social IoT Dataset*, others from external.

8.6.3 Social Internet of Things Knowledgebase Enrichment

In the instantiation process, the following enrichment are made to the Social IoT Data:

Device Entities

Devices are enriched solely from within the Social IoT Dataset. Practically, this looks like the numerical values for Device Types are replaced with string descriptors to improve readability of the results. These descriptors are:

Device Type Code	Device Type String	Population Ownership Rate %
1	Smartphone	91%
2	Car	55%
3	Tablet	40%
4	Smart Fitness	22%
5	Smart Watch	5 %
6	Personal Computer	84%
7	Printer	53%
8	Home Sensors	15 %
9	Point Of Interest	Public Infrastructure
10	Environment And Weather	Public Infrastructure
11	Transportation	Public Infrastructure
12	Indicator	Public Infrastructure
13	Garbage Truck	Public Infrastructure
14	Street Light	Public Infrastructure
15	Parking	Public Infrastructure
16	Alarms	Public Infrastructure

Further, equivalencies are made between certain *Services* listed in the *Social IoT Dataset* and the properties of a *device* in the knowledgebase. These are:

Social IoT Service	Number Code	Equivalent Vector
People Presence	4	Sight (hasCamera)
Environment	5	Sound (hasMicrophone)
Time	2	Time (hasClock)
Locations	1	Location (hasLocator)

The enrichment made to the *device* entities are limited to improving readability of results.

Persona Entities

There was limited information within the *Social IoT Dataset* that could be used to create a population with natural variance. For that reason, some randomly generated values were used to simulate the population variance, particularly regarding the vulnerability to compromise.

The first enrichment is cosmetic, with random names being generated and associated with the UserID to improve the readability of the Knowledgebase. This is achieved using the Python *Names* package to randomly generate names within the *Make Triples for Persona* function:

```
def makeTriplesForPersona(UserID, nameDict):

  # Intervening material omitted here for space

  randomName = names.get_full_name() #generates a random name for
    the user.

  #Ensure a unique UID to Name mapping.
  if UserID in nameDict.keys():
    randomName = nameDict[UserID]

  else:
    while randomName in nameDict.values():
      randomName = names.get_full_name()
```

The second was the random assignment of vulnerability to Personas. The intent of assigning vulnerabilities' is to simulate varying levels of personal privacy awareness in the population, and provide a way that the knowledgebase can be used to identify where changes to privacy policies need to be made. For example, someone who *exposes location* could represent someone who has very lax location privacy settings on their smartphone, where someone who does not *expose location* may have strict privacy setting preventing compromise. The process for deriving actual assessments of exposure from device settings is outside of the scope of this work, but when completed in future work, will be able to

easily populate into the PIANO ontology where the randomly generated values currently reside. The vulnerabilities are determined within the *MakeTriplesForPersona* function:

```
def makeTriplesForPersona(UserID, nameDict):

# Code here removed for space in thesis document

for i in range (0, 5): # loop range to stand in for each of the Persona Vulnerability dimensions of (Identity, Action, Time, Location & Motive)

    randomBool = bool(random.getrandbits(1)) # if True, the "display" of vulerablility will be instantiated, if False, it will not.

    if (i == 0):
      if randomBool == False:
        id_Bool = "privacy:exposesIdentity \"false\"^^xsd:boolean ;"
      else:
        id_Bool = "privacy:exposesIdentity \"true\"^^xsd:boolean ;"
    else:
      pass


    if (i == 1):
      if randomBool == False:
        act_Bool = "privacy:exposesAction \"false\"^^xsd:boolean ;"
      else:
        act_Bool = "privacy:exposesAction \"true\"^^xsd:boolean ;"
    else:
      pass

    if (i == 2):
      if randomBool == False:
        ti_Bool = "privacy:exposesTime \"false\"^^xsd:boolean ;"
      else:
        ti_Bool = "privacy:exposesTime \"true\"^^xsd:boolean ;"
    else:
      pass

    if (i == 3):
      if randomBool == False:
        lo_Bool = "privacy:exposesLocation \"false\"^^xsd:boolean ;"
      else:
        lo_Bool = "privacy:exposesLocation \"true\"^^xsd:boolean ;"
    else:
      pass

    if (i == 4):
      if randomBool == False:
        mo_Bool = "privacy:exposesMotive \"false\"^^xsd:boolean ."
      else:
        mo_Bool = "privacy:exposesMotive \"true\"^^xsd:boolean ."
    else:
```

```
48          pass
```

The enrichment of the Persona entity is intended to simulate natural variance of privacy awareness within the population, and provide a mechanism for data derived from the analysis of an individual's privacy settings to be inserted into the knowledgebase as part of future work.

8.6.4 Social Internet of Things Knowledegebase Automated Enhancement

A key element of the Social IoT Knowledgebase is the implementation of semi-automated *weak ontology learning* to the Social IoT Knowledgebase. In practice, this means that provided a researcher is able to instantiate the Knowledgebase as described in sections 8.6.2, 8.6.3 and 8.6.3, the ontology learning process from that point onward should require no further interaction beyond initiating the relevant python script.

The 'learning' process works in several stages, described below:

Determine Collection Vectors The collection vectors a device is collecting is derived from the device properties. Where a device has a camera (denoted by the RDF property sense:hasCamera *true*) it is assessed to be using the camera as a collection vector. The limitation of this approach is an 'always-on' view of the collection vectors. The 'always-on' approach is deliberate as it forms the *worst case* scenario for the device owner, but future work will aim to add modifiers to the device to reflect privacy controls and improve the veracity of the simulation.

To determine the collection vectors, the Social IoT Knowledgebase instantiated in the local *Stardog* instance is queried using the SPARQL semantic query language, with new relations (knowledge) being *consturcted* as the result of the query. The *DetermineCollectionVectors* Python function controls this task. A subsection of this function demonstrating how a device is determined to be collecting the *Sight* vector is below:

```
1
2  ef determineCollectionVectors(connection_details, database_name):
3    #This is a function that runs constructor queries against the
       knowledgebase to 'discover' new knowledge.
4    #Here, based on the device features derived in the by the device
       dict made from the social IOT dataset
5    #we are able to determine which collection vectors are available
       to each device.
6
7    conn = stardog.Connection(database_name, **connection_details)
8    conn.begin()
9
10
```

```
  #For Sight

  sightQuery = """
    PREFIX sense: <https://github.com/osullik/IoT-Privacy/blob/main
    /senses.ttl>
    PREFIX social_IOT_KB: <https://github.com/osullik/IoT-Privacy/
    blob/main/social_IOT_KB.ttl>

    CONSTRUCT {
        ?Device sense:collects sense:Sight
    }
    WHERE{
        ?Device sense:hasCamera true
    }
    """
  sightGraph = conn.graph(sightQuery)
```

For clarity, the SPARQL query here is:

```
    PREFIX sense: <https://github.com/osullik/IoT-Privacy/blob/main
    /senses.ttl>
    PREFIX social_IOT_KB: <https://github.com/osullik/IoT-Privacy/
    blob/main/social_IOT_KB.ttl>

    CONSTRUCT {
        ?Device sense:collects sense:Sight
    }
    WHERE{
        ?Device sense:hasCamera true
    }
```

The query effectively is determining which devices have the necessary equipment to collect sight (in this case, a camera) and using that *fact* to *assert* the next logical step - that a camera can "see" things. It creates one of these assertions for every result of the initial query, effectively "building" the relations in the Social IoT Knowledgebase Automatically. By the end of the *determineCollectionVectors* python function, knowledge has been added to the knowledgebase about what devices are able to collect what modalities.

Determine Persona Compromise Determining Persona Compromise is a process that automatically generates the internal *ontology bridge* from the collecting *device* to the vulnerable *persona*. It occurs within the *determinePersonaCompromise* python function and similarly uses the results of SPARQL queries to construct 'new' knowledge - which persona is vulnerable to privacy compromise based on the alignment of their personal vulnerability and their device's available collection vectors. An extract of the process of creating this new knowledge is below:

```
def determinePersonaCompromise(connection_details, database_name):

  conn = stardog.Connection(database_name, **connection_details)
  conn.begin()

```

```
### FOR SIGHT

## Sight to Identity (Sees Who)
seesWhoQuery = """
  CONSTRUCT{
    ?Device sense:seesWho ?Persona
    }
  WHERE{
    ?Device sense:collects sense:Sight .
    ?Persona privacy:exposesIdentity true .
    ?Device sense:deviceUser ?UID .
    ?Persona privacy:personaID ?UID
    }
"""
seesWhoGraph = conn.graph(seesWhoQuery)
```

For clarity, the SPARQL query here is:

```
  CONSTRUCT{
    ?Device sense:seesWho ?Persona
    }
  WHERE{
    ?Device sense:collects sense:Sight .
    ?Persona privacy:exposesIdentity true .
    ?Device sense:deviceUser ?UID .
    ?Persona privacy:personaID ?UID
    }
```

In plain language, the Knowledgebase will create the vulnerability relationship *Device Sees a Persona* when The device is collecting in the Sight Modality, The user is exposing their identity in a way that is vulnerable to collection by sight and the user is in possession of the device. There are a number of other possible relationships here, determined by a confluence of factors, highlighted in the table below (Where X are the vulnerabilities (and associated interrogative dimensions used to derive them), Y are the collection Vectors and the populated cells are the intersections where compromises are determined to be possible:

	Identity	Action	Time	Location	Motive
	WHO	WHAT	WHEN	WHERE	WHY
Sight	seesWho	seesWhat			seesWhy
Sound	hearsWho	hearsWhat			hearsWhy
Time			occursWhen		
Location				locatesWhere	

Using the above table to determine the points of compromise, the Social IoT Knowledgebase is able to automatically determine who is at risk of compromise by their personal device.

Determine Privacy Threats The third major 'learning' element of the Social IoT Knowledgebase is the process of determining what the privacy threats to a persona actually are. The python function *determinePrivacyImpacts* determines the privacy threats by determining which personas have compromise patterns that match a particular privacy threat, and then creating a *threatens* relationship between the persona and the privacy type. By example, the following Python code snippet illustrates how a threat to the *Privacy of the Person* is identified using a SPARQL query to construct the "new" knowledge when certain criteria are met:

```
1
2  def determinePrivacyImpacts(connection_details, database_name):
3    #The purpose of this function is to determine which privacy
       impacts are present for a given device - vector - persona
       grouping
4    # it accepts as an input the stardog endpoint details, and uses
       those to connect to the existing stardog instance
5    # it queries the instance to construct relationships where the
       appropriate conditions for a privacy risk are met (See Readme "
       Intersection" section)
6    # it returns no value but produces a .ttl file of the results,
       which it also populates into the database.
7
8    conn = stardog.Connection(database_name, **connection_details)
9    conn.begin()
10
11   ##For Personal Privacy
12
13   pp_Query = """
14     CONSTRUCT{
15         ?Persona privacy:threatens privacy:PersonalPrivacy
16     }
17     WHERE{
18         ?Device sense:deviceUser ?UID .
19         ?Persona privacy:personaID ?UID .
20         ?Device sense:seesWho ?Persona .
21         ?Device sense:hearsWho ?Persona
22         }
23     """
24   pp_Graph = conn.graph(pp_Query)
25
26   # Some code omitted here for space
27
28     ## For Thoughts and Feelings Privacy
29
30   tf_Query = """
31     CONSTRUCT{
32         ?Persona privacy:threatens privacy:ThoughtAndFeelingPrivacy
33     }
34     WHERE{
35         ?Device sense:deviceUser ?UID .
36         ?Persona privacy:personaID ?UID .
37         ?Device sense:seesWhat ?Persona .
38         ?Device sense:hearsWhat ?Persona .
39
40         {?Device sense:seesWho ?Persona}
```

```
    UNION
        {?Device sense:seesWhy ?Persona}
    UNION
        {?Device sense:hearsWho ?Persona}
    UNION
        {?Device sense:hearsWhy ?Persona}
    }
    """
```

For clarity, the two SPARQL queries here are:

```

    CONSTRUCT{
        ?Persona privacy:threatens privacy:PersonalPrivacy
    }
    WHERE{
        ?Device sense:deviceUser ?UID .
        ?Persona privacy:personaID ?UID .
        ?Device sense:seesWho ?Persona .
        ?Device sense:hearsWho ?Persona
        }


    CONSTRUCT{
        ?Persona privacy:threatens privacy:ThoughtAndFeelingPrivacy
    }
    WHERE{
        ?Device sense:deviceUser ?UID .
        ?Persona privacy:personaID ?UID .
        ?Device sense:seesWhat ?Persona .
        ?Device sense:hearsWhat ?Persona .

        {?Device sense:seesWho ?Persona}
    UNION
        {?Device sense:seesWhy ?Persona}
    UNION
        {?Device sense:hearsWho ?Persona}
    UNION
        {?Device sense:hearsWhy ?Persona}
    }
```

The First Query will assign a 'threatens' relationship between a persona and the *PersonalPrivacy* privacy type where a persona has **both** the sight and sound of their identity compromised. In the context of this ontology, this means that a device needs to be collecting with both a camera and microphone, and the user needs to be exposing their identity in a way that can both be heard and seen by their device, for example - making a video call on a smartphone.

The second query highlights greater complexity in determining when privacy is threatened. The privacy of *'thoughts and feelings'* is only threatened when A user has a device, that device is able to 'see' a users vulnerable actions and 'hear' a users vulnerable actions, and then needs to also meet **one** of the following criteria: see a users vulnerable identity, see a user's motive, hear a user's identity,

hear a user's motive. Making assessments of motive is extremely difficult, and is largely included here as a proof-of-concept. In future inferences about a user's motives could be derived from what they were saying or doing in the lead up to an event occurring, but even then - really knowing *why* anyone does anything is very difficult to achieve.

Much like the compromise matrix, there is a threatens matrix in the table below, however to read this it should be noted that for it to be *true* in the knowledgebase, all of the populated cells along the x dimension of a table need to be *true*.

	Identity	Action	Time	Location	Motive
	WHO	WHAT	WHEN	WHERE	WHY
Personal Privacy	Sight and Sound				
Behaviour and Action	Sight	Sight			
Communication	Sound	Sound	Time		
Data and Image					
Thoughts and Feeling	Sight or Sound	Sight and Sound			Sight or Sound
Location and Space	Sight or Sound			locatesWhere	
Association	Sight or Sound		Time	Location	

Where there are "or" statements in the table, it indicates that a particular facet of a persona may be vulnerable in multiple ways, for example being able to identify who someone is by a picture of their face or a recording of their voice.

The final step in the 'learning' for the knowledgebase in this way uses the knowledge generated about what a device is able to collect and who can be threatened by which devices to determine an overall assessment of which types of privacy are threatened by the combination of the user's current privacy practices and device's collection ability.

8.6.5 Summary of Social IoT Knowledgebase

The process of getting from "devices" and "personas" with their respective concept properties to an assessment of privacy vulnerability is achieved in a semi-automated fashion and reflects a significant step towards employing *ontology learning* to probe the complexity of where the IoT and Privacy domains interact, allowing us to start to quantify the risks and, in time, move towards an endstate where fully automated privacy impact assessment for users of IoT networks will be possible.

9 Evaluation

There is a substantial body of literature regarding ontology evaluation[6], [7], [24], [69], [72], [73]. Given time and ethics constraints, review by domain experts was not possible, so a multi-methodical approach to evaluation is used instead.

9.1 Evaluation Approach

The evaluation of PIANO and the Social IoT Knowledgebase has been conducted in the following ways:

1. Ongoing Quality Assurance, testing and refactoring as part of the *Agilitology* approach to ontology development.
2. Generation of summary statistics for PIANO and The Social IoT Knowledgebase
3. Tests for completeness of the representation of the usecase
4. An assessment of the utility of the Knowledgebase to addressing the business problem
5. An assessment of compliance with *Gruber's Ontology Principles*

9.2 Evaluation Results

9.2.1 Key Statistics

Namespace	Triples	Concepts	Properties	Relations
privacy	74	9	7	1
sense	110	6	6	10

Given that the dataset contains the following:

	Count	Comment
Users	4000	Mix of Public and Private
Devices	16216	14,000 Private, 1616 Public
Device Types	16	8 Private, 8 Public, Allocated by % to user
Services	7	Assigned to Device Type. Applications excluded.

The order of magnitude for **number of triples** for the Social IoT KB is substantial. A Total of **295,518** triples are used to instantiate the *Social IoT Knowledgebase*. The large number of triples here is not so much an indicator of verbosity as it is an indicator that there are over 20,000 entities represented in the Knowledgebase, each with their own properties and relations.

A deliberate decision was made to simplify queries so that they did not require the use of inferential reasoning, increasing the number of triples required. The decision is because even the number of triples required to represent the Social IoT Knowledgebase does not pose an issue in terms of memory usage, but the use of inferential reasoning did cause a considerable degradation in the time taken to execute queries.

Overall, the low number of concepts, properties and relations reinforces the intent of the ontology design to be *lightweight*, and further reflects the goal that this be a *minimally viable ontology*. The ability to keep these numbers low reflects positively on the veracity of the *Agilitology* approach to build lightweight ontologies, supporting the assessment that **it is a viable method for building lightweight ontologies**, partially answering the research question.

9.2.2 Accuracy of Representation

The accuracy of representation is important, because if the knowledgebase does not reflect reality, how can we be sure that we any knowledge derived from it is real? To that end, given the key information we know about the Social IoT dataset:

	Count	Comment
Users	4000	Mix of Public and Private
Devices	16216	14,000 Private, 1616 Public
Device Types	16	8 Private, 8 Public, Allocated by % to user
Services	7	Assigned to Device Type. Applications excluded.

We can use SPARQL queries to verify that the same information is reflected in the Social IoT Knowledgebase

Are all 4000 users represented in the KB?

```
SELECT (COUNT(?User) as ?UserCount)
WHERE{
    ?Persona privacy:personaID ?User
}
```

Returns 4001. On closer inspection, the dataset does not include "user0" as a user. User0 is reflective of a systemUser on behalf of all public devices and as

such has over a thousand public devices associated with their name. There is no impact to the accuracy of the knowledgebase from this.

Are all 16216 devices listed?

```
SELECT (COUNT(?Device) as ?DeviceCount)
WHERE{
    ?Device rdf:type sense:Device
}
```

Returns: 16215. After reviewing the documentation and the relevant data, there continues to be one device missing from the Knowledgebase. The missing device is *Device Number 1*. Review of the data pushed into the KB indicates that the user associated with this data is still present. On review of the relevant python functions, a likely culprit has been identified (shown in python snipped below, but is unable to be removed at this stage as it inflects a breaking change on the remainder of the code. It will be remediated as part of future work:

```
  # Generates the dictionary of key information from the dataset to
     feed the knowledgeBase creation.
  #Note that the device Type is used as the Key in the servicesDict
     as the services are similar across device type.
  count = 0 # a workaround to remove the header information, that
    was breaking the RDF Triple generation downstream by skipping
    row 0.
  for row in object_description:
    if count == 0:
      pass
    else:
      devicesDict[row[0]] = {"deviceType": deviceTypeDict[row[2]] ,
     "userID":row[1], "deviceServices":servicesDict[row[2]]}
    count+= 1
```

Skipping the first row is ***likely*** the cause of missing this entry, however it has a negligible overall impact at this time.

The third aspect of accuracy of representation we can judge is whether the distribution of devices in the Social IoT Knowledgebase match those cited in the dataset documentation [27].

The SPARQL Query is:

```
    # Determine what device type is most prolific
SELECT ?DeviceType (COUNT(?Name) as ?UserCount)
WHERE{
    ?Device sense:deviceType ?DeviceType .
    ?Device sense:deviceUser ?UID .
    ?Persona privacy:personaID ?UID .
    ?Persona privacy:personaName ?Name
}
GROUP BY ?DeviceType
ORDER BY DESC(?UserCount)
```

Device Type	Number in KB	% of Total in KB	Population %
Smartphone	3639	91%	91%
Car	2200	55%	55%
Tablet	1600	40%	40%
Smart Fitness	880	22%	22%
Smart Watch	200	5 %	5 %
Personal Computer	3360	84%	84%
Printer	2120	53%	53%
Home Sensors	600	15 %	15 %
Point Of Interest	95	6%	Not Given
Environment And Weather	140	9%	Not Given
Transportation	143	9%	Not Given
Indicator	10	1%	Not Given
Garbage Truck	7	1%	Not Given
Street Light	506	31%	Not Given
Parking	677	42%	Not Given
Alarms	38	24%	Not Given

As shown in the table, the proportional representation of devices in the Knowledgebase is reflective of the Dataset they are derived from. To the extent that the dataset can be confirmed as successfully instantiated into the Knowledgebase, it is **confirmed**. The confirmation supports the veracity of the current ingestion pipeline and transformation into RDF triples, less the single missing *device* which is accounted for.

9.2.3 Utility to Business Problem (Action Research Output)

To determine if PIANO and the Social IoT Knowledgebase is ***Useful***, action research (and the DSRM Approach encapsulated within the *Agilitology* approach requires an evaluation against realistic business problems for utility.

Based on the assessment of the researcher, the following competency questions were determined as being of **LIKELY** interest to privacy researchers, regarding the interactions of IoT Devices and Individual's privacy.

1. What Privacy Risks is an Individual being threatened by?
2. Which user has the most threats to their privacy?
3. Which type of privacy threat is the most common?
4. Which device Type threatens the most privacy types?
5. Who is vulnerable to a given privacy threat?

6. How many threats is a each person vulnerable to?
7. How many Vectors is each person collected on by?
8. How many privacy Exposures is each person allowing?

What Privacy Risks is an Individual being threatened by

```
    # What privacy risks is an individual vulnerable to?

SELECT ?Name ?PrivacyRisk
WHERE{
    ?Persona privacy:threatens ?PrivacyRisk .
    ?Persona privacy:personaName ?Name
}
```

Results are in the figure 6 below. Key to note here is that we can get a list of each type of threatened privacy. This would allow a privacy analyst to understand the specific risks being faced by an individual when helping them to design a personal privacy plan.

Name	PrivacyRisk
"Constance Mcgee"	privacy:PersonalPrivacy
"Constance Mcgee"	privacy:BehaviourAndActionPrivacy
"Constance Mcgee"	privacy:CommunicationPrivacy
"Constance Mcgee"	privacy:ThoughtAndFeelingPrivacy
"Constance Mcgee"	privacy:LocationAndSpacePrivacy
"Constance Mcgee"	privacy:AssociatonPrivacy
"Margaret Siciliano"	privacy:PersonalPrivacy
"Margaret Siciliano"	privacy:BehaviourAndActionPrivacy
"Margaret Siciliano"	privacy:CommunicationPrivacy
"Margaret Siciliano"	privacy:ThoughtAndFeelingPrivacy
"Margaret Siciliano"	privacy:LocationAndSpacePrivacy
"Margaret Siciliano"	privacy:AssociatonPrivacy

Figure 6: Results of Business Competency Question 1

Which user has the most threats to their privacy?

```
        # Determine who has the most privacy Violations
    SELECT ?Name (COUNT (?PrivacyRisk) as ?RiskCount)
    WHERE{
        ?Persona privacy:threatens ?PrivacyRisk .
        ?Persona privacy:personaName ?Name
    }
    GROUP BY ?Name
```

Results are in the figure 7. Noting that I removed the *order By* clause from the above to demonstrate differing numbers, it is apparent to see that some

individuals, based on their combination of exposed attributes and the vectors being collected by their devices are having all six types of encoded privacy type threatened. As a privacy advisor, we could use this tool to triage who is most at risk, and assign effort to assist those people first.

Name	RiskCount
"Maria Rogers"	6
"Garrett Rickman"	6
"Mary Decambra"	6
"Judith Richards"	6
"Corrine Olaughlin"	1
"Jose Lucius"	6
"Malinda Rafter"	1
"Leonard Zapata"	6
"Annette Curly"	6
"Jerome Vasquez"	3
"Kaleigh Garfield"	6
"Earlene Martinetto"	3
"Theresa Bernier"	6
"Eric Johnson"	6

Figure 7: Results of Business Competency Question 2

Which type of privacy threat is the most common?

```
# Determine which type of privacy violation is the most common.
SELECT ?PrivacyRisk (COUNT (?Name) as ?NameCount)
WHERE{
    ?Persona privacy:threatens ?PrivacyRisk .
    ?Persona privacy:personaName ?Name
    }
GROUP BY ?PrivacyRisk
ORDER BY DESC(?NameCount)
```

Results are in the figure 8 below. However, based on these results, as a privacy

analyst I can immediately see that this population is, for whatever reason, less vulnerable to having their privacy of communication threatened than their personal privacy. If this was a sample from an organisation, we could look to implement organisational policies to specifically reduce risk associated with the most prolific privacy threat to improve the overall privacy climate based on this knowledge.

PrivacyRisk	NameCount
privacy:PersonalPrivacy	3509
privacy:ThoughtAndFeelingPriva...	3469
privacy:BehaviourAndActionPriv...	3222
privacy:AssociatonPrivacy	2878
privacy:LocationAndSpacePrivacy	2878
privacy:CommunicationPrivacy	2877

Figure 8: Results of Business Competency Question 3

Which device Type threatens the most privacy types?

```
# Determine which Device type has the most associated
PrivacyImpacts (Note: one device may pose multiple risks)
SELECT ?DeviceType (COUNT(?PrivacyRisk ) as ?RiskCount)
WHERE{
    ?Device sense:deviceType ?DeviceType .
    ?Device sense:deviceUser ?UID .
    ?Persona privacy:personaID ?UID .
    ?Persona privacy:threatens ?PrivacyRisk
}
GROUP BY ?DeviceType
ORDER BY DESC(?RiskCount)
```

Results are in the figure 9 below. Unsurprisingly, smartphones and personal computers are involved in the largest number of privacy threats. But oddly, printers also score very highly. If I was conducting a privacy risk assessment for the IT department of a company, I might be very interested to know that the unsuspecting printers we are considering procuring pose a significant privacy risk to the workforce. (Note that these numbers are higher than the actual number of devices as a single smartphone may threaten up to six discrete privacy types)

Who is vulnerable to a given privacy threat?

```
# Determine who is vulnerable to a given privacy violation
SELECT DISTINCT ?Name
WHERE{
    ?Device sense:deviceUser ?UID .
    ?Persona privacy:personaID ?UID .
```

DeviceType	RiskCount
"Smartphone"	17609
"PersonalComputer"	16192
"Car"	11143
"Printer"	10644
"Tablet"	8264
"SmartFitness"	4505
"HomeSensors"	3079
"Parking"	2708
"StreetLight"	2024
"SmartWatch"	1030
"Transportation"	572
"EnvironmentAndWeather"	560
"PointOfInterest"	380
"Alarms"	152
"Indicator"	40
"GarbageTruck"	28

Figure 9: Results of Business Competency Question 4

```
        ?Persona privacy:personaName ?Name .
        ?Persona privacy:threatens privacy:PersonalPrivacy
    }
```

Results are in the figure 10 below. As a privacy Analyst, if a query came in about which members of my organisation were having a particular aspect of their privacy threatened, this would allow me to quickly identify those people, and perhaps allow me to target educational material on how to reduce their vulnerability, increasing the overall resilience of the organisation.

How many Vectors is each person collected on by?

```
    # Determine how many vectors each person is vulnerable to, and
```

Name
"Constance Mcgee"
"Margaret Siciliano"
"Samuel Zachary"
"Angela Birkline"
"Erica Christensen"
"Virginia Gibeau"
"Michael Lewis"
"Era Brock"
"Willie Elliott"
"Andrew Byrd"
"Matthew Manke"
"Jason Oshaughnessy"
"Darlene Slaugh"
"Donna Peterson"
"Douglas Stanley"
"Juan Williams"

Figure 10: Results of Business Competency Question 5

```
    how many of each vector they are at risk to
2   SELECT ?Name ?NumberofDevices ?CollectionVectors ?SightVectors
    ?SoundVectors ?TimeVectors ?LocationVectors
3       WHERE{
4
5           {
6               SELECT (COUNT(sense:Sight) as ?SightVectors) ?Name
7
8               WHERE{
9               ?Persona privacy:personaID ?UID .
10              ?Device sense:deviceUser ?UID .
11              ?Persona privacy:personaName ?Name .
12              ?Device sense:collects sense:Sight
13              }
14              GROUP BY ?Name
```

```
15              }
16
17              {
18                  SELECT (COUNT(sense:Sound) as ?SoundVectors) ?Name
19
20                  WHERE{
21                  ?Persona privacy:personaID ?UID .
22                  ?Device sense:deviceUser ?UID .
23                  ?Persona privacy:personaName ?Name .
24                  ?Device sense:collects sense:Sound
25                  }
26                  GROUP BY ?Name
27              }
28
29                      {
30                  SELECT (COUNT(sense:Time) as ?TimeVectors) ?Name
31
32                  WHERE{
33                  ?Persona privacy:personaID ?UID .
34                  ?Device sense:deviceUser ?UID .
35                  ?Persona privacy:personaName ?Name .
36                  ?Device sense:collects sense:Time
37                  }
38                  GROUP BY ?Name
39              }
40
41                      {
42                  SELECT (COUNT(sense:Location) as ?LocationVectors)
    ?Name
43
44                  WHERE{
45                  ?Persona privacy:personaID ?UID .
46                  ?Device sense:deviceUser ?UID .
47                  ?Persona privacy:personaName ?Name .
48                  ?Device sense:collects sense:Location
49                  }
50                  GROUP BY ?Name
51              }
52
53                     {
54                  SELECT (COUNT(?Vector) as ?CollectionVectors) ?Name
55
56                  WHERE{
57                  ?Persona privacy:personaID ?UID .
58                  ?Device sense:deviceUser ?UID .
59                  ?Persona privacy:personaName ?Name .
60                  ?Device sense:collects ?Vector
61                  }
62                  GROUP BY ?Name
63                  ORDER BY DESC(?CollectionVectors)
64              }
65
66                              {
67                  SELECT (COUNT(?Device) as ?NumberofDevices) ?Name
68
69                  WHERE{
70                  ?Persona privacy:personaID ?UID .
```

```
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                }
                GROUP BY ?Name
            }
        }
```

Results are in the figure 11 below. Noting I removed the ordering of risks to demonstrate the spread of results we can immediately deduce a few things from the results. First, that the number of collection vectors absolutely correlates to the number of devices owned. And the second is that some people are exceptions to this correlation. For example, *Julie Mcdonald* here owns four devices, but is only vulnerable to six vectors. People like Julie could be identified as *privacy champions* within the organisation to try and lead privacy-conscious revolution, or at the very least should be a model of how to reduce risk, based on her personal device configuration.

Name	NumberofDevices	CollectionVectors	SightVectors	SoundVectors	TimeVectors	LocationVectors
"Maria Rogers"	5	10	3	2	3	2
"Richard Fincher"	1	4	1	1	1	1
"James Rauch"	4	10	3	2	3	2
"Mary Vida"	6	11	3	3	3	2
"Garrett Rickman"	6	15	4	4	4	3
"Billy Adams"	4	10	3	2	3	2
"Tommy Mcadoo"	3	10	3	2	3	2
"Jessica Hodgson"	3	10	3	2	3	2
"Joann Dolan"	4	11	3	3	3	2
"Jason Vanhofwegen"	4	10	3	2	3	2
"Joseph Gulbrandsen"	4	14	4	3	4	3
"Christopher Hilchey"	5	13	4	3	4	2
"Lynn Whitt"	4	10	3	2	3	2
"Julie Mcdonald"	4	6	2	1	2	1
"James Hunter"	4	14	4	3	4	3
"Wanda Barker"	3	10	3	2	3	2

Figure 11: Results of Business Competency Question 6

How many privacy Exposures is each person allowing?

```
# Determine how many Vulnerabilites each person is exposing, and how many of each type
SELECT ?Name ?NumberofDevices ?CompromiseVulnerabilities ? IdentityVulnerabilites ?ActionVulnerabilites ? TimeVulnerabilities ?LocationVulnerabilities ? MotiveVulnerabilities
    WHERE{

               {
            SELECT (COUNT(?Persona) as ? CompromiseVulnerabilities) ?Name

            WHERE{
            ?Persona privacy:personaID ?UID .
            ?Device sense:deviceUser ?UID .
            ?Persona privacy:personaName ?Name .

            {?Persona privacy:exposesIdentity true}
            UNION
            {?Persona privacy:exposesAction true}
            UNION
            {?Persona privacy:exposesTime true}
            UNION
```

```
                {?Persona privacy:exposesLocation true}
                UNION
                {?Persona privacy:exposesMotive true}
                }
                GROUP BY ?Name
                ORDER BY DESC(?CompromiseVulnerabilities)
            }

            {
                SELECT (COUNT(privacy:exposesIdentity) as ?
    IdentityVulnerabilites) ?Name

                WHERE{
                ?Persona privacy:personaID ?UID .
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                ?Persona privacy:exposesIdentity true
                }
                GROUP BY ?Name
            }

            {
                SELECT (COUNT(privacy:exposesAction) as ?
    ActionVulnerabilites) ?Name

                WHERE{
                ?Persona privacy:personaID ?UID .
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                ?Persona privacy:exposesAction true
                }
                GROUP BY ?Name
            }

            {
                SELECT (COUNT(privacy:exposesTime) as ?
    TimeVulnerabilities) ?Name

                WHERE{
                ?Persona privacy:personaID ?UID .
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                ?Persona privacy:exposesTime true
                }
                GROUP BY ?Name
            }

            {
                SELECT (COUNT(privacy:exposesLocation) as ?
    LocationVulnerabilities) ?Name

                WHERE{
                ?Persona privacy:personaID ?UID .
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                ?Persona privacy:exposesLocation true
                }
```

```
                GROUP BY ?Name
            }

            {
                SELECT (COUNT(privacy:exposesMotive) as ?
    MotiveVulnerabilities) ?Name

                WHERE{
                ?Persona privacy:personaID ?UID .
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                ?Persona privacy:exposesMotive true
                }
                GROUP BY ?Name
            }

                    {
                SELECT (COUNT(?Device) as ?NumberofDevices) ?Name

                WHERE{
                ?Persona privacy:personaID ?UID .
                ?Device sense:deviceUser ?UID .
                ?Persona privacy:personaName ?Name .
                }
                GROUP BY ?Name
            }

        }
```

Results are in the figure 12 below. Similar to the previous report, a privacy analyst would be able to use this report to calculate an organisation's overall privacy health for each exposure risk. If there is an unusually high number of *identity* vulnerabilities for example, it could justify an organisational education campaign to improve awareness about identity exposure and start taking steps to reduce it.

Name	NumberofDevices	CompromiseVulnerabilit...	IdentityVulnerabilites	ActionVulnerabilites	TimeVulnerabilities	LocationVulnerabilities	MotiveVulnerabilities
"Maria Rogers"	5	25	5	5	5	5	5
"James Rauch"	4	20	4	4	4	4	4
"Mary Vida"	6	30	6	6	6	6	6
"Garrett Rickman"	6	30	6	6	6	6	6
"Jason Vanhofwegen"	4	20	4	4	4	4	4
"Christopher Hilchey"	5	25	5	5	5	5	5
"Lynn Whitt"	4	20	4	4	4	4	4
"Julie Mcdonald"	4	20	4	4	4	4	4
"James Hunter"	4	20	4	4	4	4	4
"Wanda Barker"	3	15	3	3	3	3	3
"Mark Ornelas"	5	25	5	5	5	5	5
"Jason Wallace"	3	15	3	3	3	3	3
"Roy Coleman"	5	25	5	5	5	5	5
"Judy Weber"	3	15	3	3	3	3	3
"Carla Kledrowski"	3	15	3	3	3	3	3
"Stanley Thompson"	4	20	4	4	4	4	4

Figure 12: Results of Business Competency Question 7

9.2.4 Compliance with Gruber's principles for ontology design

Gruber's principles of ontology design are *Clarity, Coherence, Extensiblity, Minimal Encoding Bias* and *Minimum Ontological Commitment*[7]. PIANO per-

forms against these criteria as follows:

Clarity. The concept, relation and property names have been specifically engineered to be descriptive. Further, each concept, relation and property contains a clear label and description. Within the comments of the ontology, sourcing is given for the definitions, whether that be the work of Finn Et. Al [13] for privacy, Marche et. al. [27] for dataset definitions or the use of commonly understood concepts (e.g. Who, What, When, Where and Why) to clearly articulate what is meant by each concept, property and relation. To the extent that one can self-evaluate clarity, the efforts made here have been **GOOD**.

Coherence. The evaluations conducted in 9.2.3 reflect logical inferences that are consistent with what is expected of a coherent ontology. Because the PIANO is built iteratively using the *Agilitology* method, the resulting construct is *lightweight* and minimally complex. Because of the minimal complexity, complex inferencing is not required and the probability of incoherence decreases significantly. Without extensive testing, I am reluctant to assess PIANO as anything higher than **SATISFACTORY**

Extensibility. PIANO is explicitly designed as an abstract ontology to sit between two ontology bridges into the granular *IoT* and *people* domains respectively. As a result, the novel components of the ontology are minimal, and also highly flexible to interaction with different ontologies. The ontology itself has been developed using the *Agilitology* approach which enforces an *iterative* approach to build a *minimally viable ontology* that is *lightweight*, with only the minimum number of concepts, properties and relations required to answer the competency questions that specify its functionality requirements. By its very develop approach, it is inherently extensible, when more requirements arise they can be added. I assess the Extensibility of PIANO as **GOOD**

Minimal Encoding Bias. The use-case driven approach and focus on producing something usable that is driven by *Agilitology* has the significant drawback that it generates a heavy encoding bias. PIANO currently only exists as RDF triples and python scripts. There is no generalised mathematical representation that could be easily implemented in other languages. Agile approaches are noted for their desire for *working software over comprehensive documentation* and the manifestation here is a **POOR** ability to minimise encoding bias.

Minimal Ontological Commitment. The *Agilitology* ensures that only the bare minimum of concepts, relations and properties that are required to support the intended knowledge are included. This is evidenced by the raw counts summarised in section 9.2.1 that show very few concepts, relations and properties are used. Overall, fewer than 200 RDF Triples are required to construct a knowledgebase of over 250,000 relations. The simplicity of the ontological structure itself is further evident in the query efficiency, with none of the test queries run taking longer than 1 second to execute on a 2014 Macbook Pro.

PIANO does a **GOOD** job minimising ontological commitment.

Summary of Gruber's Principles Based on subjective assessment against Gruber's principles, the overall assessment of PIANO is **GOOD**. The component assessments on a (Poor - Satisfactory - Good) scale are summarised in the below table:

Gruber's Principle	Self-Assessment
Clarity	**GOOD**
Coherence	**SATISFACTORY**
Extensibility	**GOOD**
Minimal Encoding Bias	**POOR**
Minimal Ontological Commitment	**GOOD**

9.3 Summary of Evaluation

Overall, the evaluation of PIANO and the Social IoT Knowledgebase indicate that its biggest weakness is a heavy encoding bias, driven by the use of the *Agilitology* approach to ontology development. However, PIANO abides by all other ontology design principles, has clear and immediate utility to addressing the business problems and has accurately represented the Social IoT Dataset in a semantic format.

10 Conclusion and Future Work

10.1 Summary of Research

This thesis set out to address t**he business problem** where there was currently no mechanism to assess the impact of IoT networks on an individual's privacy. The Evaluation presented in section 9.2.3 highlight the immediate utility to privacy analysts of using PIANO to instantiate their own Knowledgebase with their own data to determine where the ability of IoT devices to collect, and the vulnerability of people to expose themselves intersect. PIANO has made significant progress in proving the viability of the semantic approach for further work in automating privacy impact assessment.

The scientific problem posed was that there was no established method to derive ontological structures that will enable Privacy impact assessments. The methodology used to generate PIANO and the Social IoT Knowledgebase is described in section 8.1.2. It uses the *Agilitology* approach to manually derive a *minimally viable ontology* that is then instantiated with IoT data and subject to

several rounds of *ontology learning* to derive new insights about the intersection of privacy and IoT devices. PIANO and the Social IoT Knowledgebase have made significant progress to developing a clear, repeatable method for further scientific research. This is made easier by the public access to the code used to develop PIANO and the Social IoT Knowledgebase on the Author's github.

The **Research question** this thesis addressed was *What is an ontology development methodology that can dynamically generate ontological structures to support privacy impact estimation for the internet of things.*

The question was decomposed into three parts. The first of which was *What is a suitable ontology development method to derive semantic relationships from semi-structured source data such as XML files PCAP files and log files?*

The first sub-question was actually not answered as part of this thesis. The key reason was that the dataset that was selected per the criteria in section 8.5 did not contain this kind of data. While the requirement to conduct ontology learning from .PCAP files and other semi-structured data sources remains open, it has been determined as irrelevant within the scope of this project and has not been pursued further.

The second research sub-question was *What is a suitable ontology development method to be used to derive semantic links between IoT device data and privacy concepts?*

Based on the results of this research, the two part answer is, first the *Agilitology* approach to ontology development, which is highly suited for operating in complex domains, like those identified at the intersection of Privacy and IoT in section 6.1. This is because it requires the researcher to *act*, and interact with the domain. That interaction stimulates a response, and that response can be observed. With enough stimulation-observation loops, an *action researcher* applying *Agilitology* will quickly be able to produce a solution that is *minimally viable* for the domain. Beyond the *Agilitology* approach itself, the method used to develop PIANO and the Social IoT Knowledgebase described in section 8.1.2 is a suitable method to derive links between privacy and IoT domains.

The third and final sub-question was *What is a suitable system architecture to enable the conduct of these two ontology learning tasks?*

While further experimentation is needed to determine if there are better solutions available, the solution employed by the researcher of this thesis is **free** (aside from the cost of the hardware). The system architecture this was built on is in the figure 13 below.

This architecture, in addition to the development methodology provides a suitable approach to develop PIANO and the Social IoT Knowledgebase. There is one point to note a likely lack of replicability. The "names" associated with

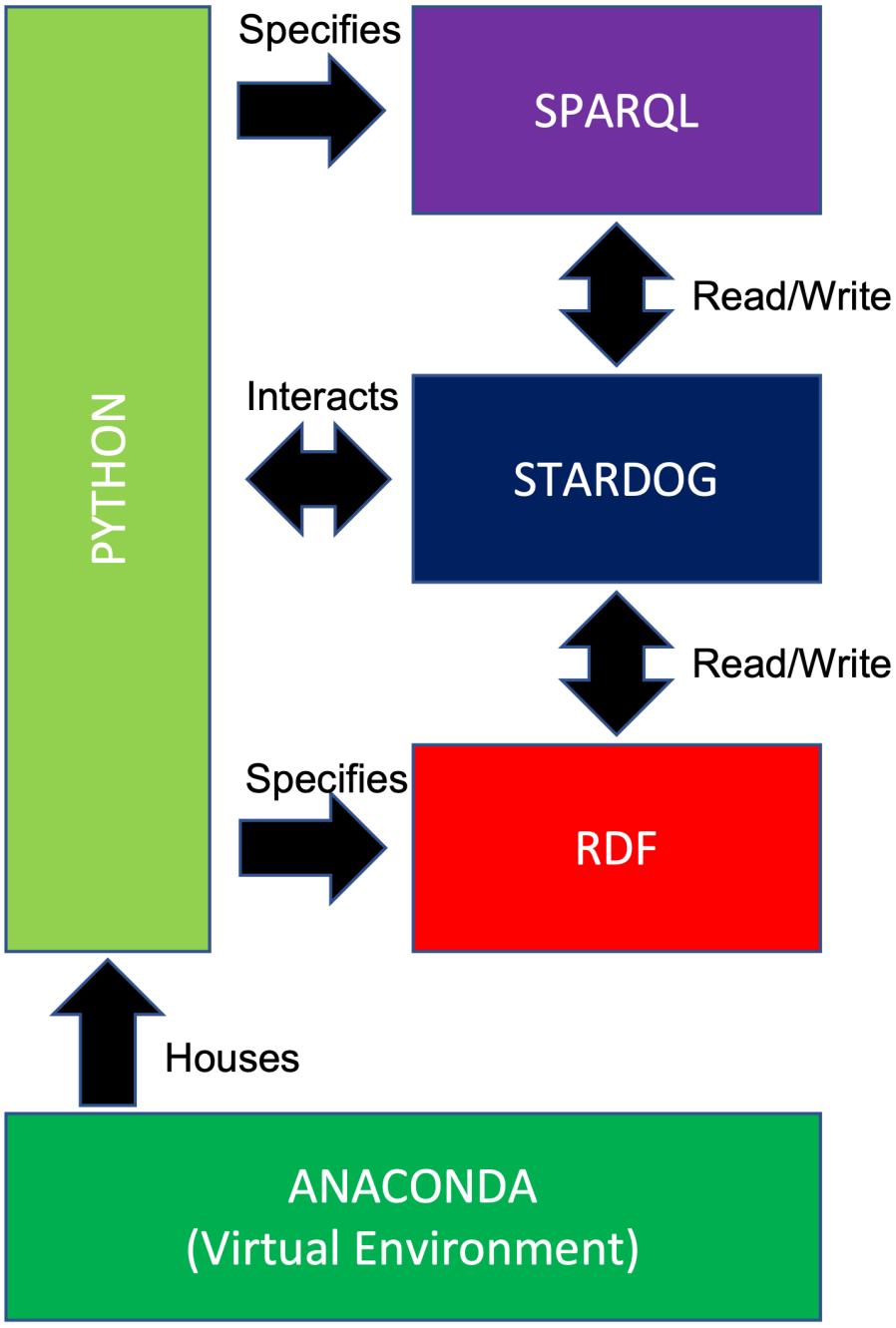

Figure 13: PIANO Tech Stack

the users are randomly generated each instantiation and so will not remain constantly aligned with a user ID. This can be remedied with minor works if it becomes a requirement. Further, the random allocation of privacy exposures will not be exactly replicable.

Overall, the thesis has addressed the research questions, and produced a substantial body of work to support the conclusion that PIANO presents a viable example of an ontology generation methodology that can meaningfully bridge the IoT and Privacy Domains.

10.2 Future Work

There are large amounts of future work noted in various sections throughout this thesis. However of particular focus should be:

1. Instantiate a second, disparate dataset to test that PIANO can generalise to other data sources (beyond the *Extract, Transform* and *load* phases described in section 8.6.2. The *Learning* aspects of the system should all work.
2. Related to the first point, minor refactoring of the code and tidying up of the git page this is stored should occur to maximise the chance that PIANO will be used in future research projects.
3. Where notional data is currently used (device features, user exposure) effort should be applied to develop mechanisms to derive real features from their appropriate granular ontologies to feed PIANO analysis.
4. PIANO should extend the *CollectionVectors* concept to account for differing levels of security controls (i.e. cameras aren't always on) to better reflect an individual's device security controls.
5. PIANO should extend to handle out-of-band collection. That is, unauthorised access or other forms of cyber security incident that would intercept *data* as well as being able to turn on cameras, location services etc.
6. Not specifically related to this project, but in general the lack of established methodology for data mining from .PCAPs and other structured network data is in need of further examination.
7. Refactoring the Ontology into an encoding-agnostic format to improve the current poor performance related to its heavy encoding bias.

10.3 Conclusion

This thesis set out to develop a methodology to develop an ontological structure that would be able to link the *privacy* and *Internet of Things* domains. The *Agilitology* approach has produced the *Privacy Impact Assessment Nexus Ontology*, or **PIANO**, and the affiliated knowledgebase *Social IoT Knowledgebase* which was built out of the *Social Internet of Things Dataset* [27].The creation of these artefacts, which have been evaluated from a number of perspectives and performed well is evidence that a suitable methodology has been developed.

10.3.1 Novel Contributions

The following novel contributions are identified as originating from this research:

1. **Privacy Types**. PIANO is the first privacy ontology that is not tied to a specific legislative system or focused exclusively on compliance. It uses a peer-reviewed privacy model, instantiated in code to give an objective, independent assessment of privacy risks.
2. **Ontology Bridging**. PIANO is an abstract ontology that is able to connect granular IoT and Person ontologies, enriching their link with privacy impact assessment information. It is generic enough to extend most ontologies from either domain, and lightweight enough that it will not be cumbersome to deploy.
3. **Social IoT Knowledgebase** The Social IoT Knowledgebase is the first attempt to link IoT data to objective, independent privacy risk assessments. It is also published on the Author's Github.
4. **Agilitology** The Agilitology approach was deployed outside of the *Cyber Security* domain for the first time, demonstrating the flexibility of this approach to solving problems in complex environments.

11 Aftermatter

11.1 Research Ethics

No ethics approval was requested or required for the conduct of this research.

11.2 Intellectual Property

All technical artifacts have been released on the Author's Github under a GNU General Public Licence version 3.

References

[1] Wei Wang, Suparna De, Ralf Toenjes, et al. "A comprehensive ontology for knowledge representation in the internet of things". In: *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE. 2012, pp. 1793–1798.

[2] Guangquan Xu, Yan Cao, Yuanyuan Ren, et al. "Network security situation awareness based on semantic ontology and user-defined rules for Internet of Things". In: *IEEE Access* 5 (2017), pp. 21046–21056.

[3] Sara Hachem, Thiago Teixeira, and Valerie Issarny. "Ontologies for the internet of things". In: *Proceedings of the 8th middleware doctoral symposium*. 2011, pp. 1–6. URL: https://hal.inria.fr/docs/00/64/21/93/PDF/IotMiddleware.pdf.

[4] Vıctor Caballero, Sergi Valbuena, David Vernet, et al. "Ontology-Defined middleware for internet of things architectures". In: *Sensors* 19.5 (2019), p. 1163.

[5] Tr Gruber. "A Translation Approach To Portable Ontology Specifications, ontology, semantic, ontology, semantic, cyber". In: *Knowledge Acquisition* 5.2 (1993), pp. 199–220. ISSN: 1042-8143. URL: https://is.muni.cz/el/phil/jaro2017/VIKMA05/um/68063749/Gruber-toward_principles.pdf.

[6] Mike Uschold and Michael Gruninger. "Ontologies: Principles, methods and applications". In: *The knowledge engineering review* 11.02 (1996), pp. 93–136. ISSN: 1469-8005. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.5917&rep=rep1&type=pdf.

[7] Thomas R Gruber. "Toward principles for the design of ontologies used for knowledge sharing". In: *International journal of human-computer studies* 43.5 (1995), pp. 907–928. ISSN: 1071-5819.

[8] Natalya F Noy and Deborah L McGuinness. *Ontology Development 101: A Guide To Developing Your First Ontology, ontology, semantic, ontology, semantic, cyber*. Report. Stanford Knowledge Systems Laboratory, 2001.

[9] Oscar Corcho, Mariano Fernandez-Lopez, and Asuncion Gomez-Perez. "Ontological engineering: What are ontologies and how can we build them?" In: *Semantic Web Services*. Nueva York: Premier Reference Source, 2007, pp. 44–70.

[10] Dave Adamy. "Electronic Warfare 101: 5G Communications". In: *Journal of Electromagnetic Dominance* (2021).

[11] Daniel J Solove. "A taxonomy of privacy". In: *U. Pa. L. Rev.* 154 (2005), p. 477. URL: https://scholarship.law.gwu.edu/cgi/viewcontent.cgi?article=2074&context=faculty_publications.

[12] Debbie VS Kasper. "The evolution (or devolution) of privacy". In: *Sociological Forum*. Vol. 20. 1. Springer. 2005, pp. 69–92.

[13] Rachel L Finn, David Wright, and Michael Friedewald. "Seven types of privacy". In: *European data protection: coming of age*. Springer, 2013, pp. 3–32.

[14] Monica Palmirani, Michele Martoni, Arianna Rossi, et al. "PrOnto: Privacy ontology for legal reasoning". In: *International Conference on Electronic Government and the Information Systems Perspective*. Springer. 2018, pp. 139–152. URL: https://cris.unibo.it/retrieve/handle/11585/648022.8/467479/POST%20PRINT%20PrOntoPrivacyOntologyForLegalReasoning.pdf.

[15] Monica Palmirani, Michele Martoni, Arianna Rossi, et al. "Legal Ontology for Modelling GDPR Concepts and Norms." In: *JURIX*. 2018, pp. 91–100. URL: https://doras.dcu.ie/23801/1/DPVCG___ODBASE.pdf.

[16] Noria Foukia, David Billard, and Eduardo Solana. "PISCES: A framework for privacy by design in IoT". In: *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE. 2016, pp. 706–713.

[17] Leo Obrst, Werner Ceusters, Inderjeet Mani, et al. "The evaluation of ontologies". In: *Semantic Web*. Springer, 2007, pp. 139–158. ISBN: 0387484361. URL: http://www.mel.nist.gov/msidlibrary/doc/eval_ontologies.pdf.

[18] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. "Methontology: from ontological art towards ontological engineering". In: (1997). URL: http://oa.upm.es/5484/1/METHONTOLOGY_.pdf.

[19] Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez, et al. "Building legal ontologies with METHONTOLOGY and WebODE". In: *Law and the semantic web*. Springer, 2005, pp. 142–157. ISBN: 3540250638.

[20] Mariano Fernández López, Asunción Gómez-Pérez, Juan Pazos Sierra, et al. "Building a chemical ontology using methontology and the ontology design environment, ontology, semantic, ontology, semantic, cyber". In: *IEEE intelligent Systems* 1 (1999), pp. 37–46. ISSN: 1541-1672.

[21] Ahlam Sawsaa and Joan Lu. "Building Information Science ontology (OIS) with Methontology and Protégé, ontology, semantic, ontology, semantic, cyber". In: *Journal of Internet Technology and Secured Transactions (JITST)* 1.3/4 (2012). ISSN: 2046-3723.

[22] Kent O'Sullivan and Benjamin Turnbull. "The cyber simulation terrain: Towards an open source cyber effects simulation ontology". In: *16th Australian Information Warfare Conference*. Edith Cowan University. Security Research Institute, Edith Cowan University, Dec. 2015, pp. 14–23. DOI: 10.4225/75/57a84e3bbefbc. URL: https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1059&context=isw.

[23] Alexander Maedche. *Ontology Learning for the Semantic Web*. 1st ed. Springer, 2002.

[24] Ahlem Chérifa Khadir, Hassina Aliane, and Ahmed Guessoum. "Ontology learning: Grand tour and challenges". In: *Computer Science Review* 39 (2021), p. 100339.

[25] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a standard process model for data mining". In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Vol. 1. Springer-Verlag London, UK. 2000. URL: https://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf.

[26] Nour Moustafa. $TON_IoT(UNSW - IoT - 20$. 2020. URL: https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i.

[27] Claudio Marche, Luigi Atzori, Virginia Pilloni, et al. "How to exploit the Social Internet of Things: Query Generation Model and Device Profiles' Dataset". In: *Computer Networks* 174 (2020), p. 107248. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2020.107248. URL: https://www.sciencedirect.com/science/article/pii/S138912861931730X.

[28] Ivan Bedini and Benjamin Nguyen. "Automatic ontology generation: State of the art". In: *PRiSM Laboratory Technical Report. University of Versailles* (2007), pp. 1–15. URL: http://bivan.free.fr/Janus/Docs/Automatic_Ontology_Generation_State_of_Art.pdf.

[29] Hamid Mousavi, Deirdre Kerr, Markus Iseli, et al. "Harvesting domain specific ontologies from text". In: *2014 IEEE International Conference on Semantic Computing*. IEEE. 2014, pp. 211–218. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.571.9674&rep=rep1&type=pdf.

[30] Mokhtaria Hacherouf, Safia Nait Bahloul, and Christophe Cruz. "Transforming XML documents to OWL ontologies: A survey". In: *Journal of Information Science* 41.2 (2015), pp. 242–259. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.841.1359&rep=rep1&type=pdf.

[31] Muhammad Shafiq, Zhihong Tian, Ali Kashif Bashir, et al. "Data mining and machine learning methods for sustainable smart cities traffic classification: A survey". In: *Sustainable Cities and Society* 60 (2020), p. 102177.

[32] Man Zhu, Zhiqiang Gao, Jeff Z Pan, et al. "TBox learning from incomplete data by inference in BelNet+". In: *Knowledge-Based Systems* 75 (2015), pp. 30–40. URL: https://knowledge-representation.org/j.z.pan/pub/ZGPZ*2015.pdf.

[33] Gerhard Wohlgenannt and Filip Minic. "Using word2vec to Build a Simple Ontology Learning System." In: *International Semantic Web Conference (Posters and Demos)*. 2016. URL: http://ceur-ws.org/Vol-1690/paper37.pdf.

[34] Biralatei Fawei, Jeff Z Pan, Martin Kollingbaum, et al. "A semi-automated ontology construction for legal question answering". In: *New Generation Computing* 37.4 (2019), pp. 453–478. URL: https://link.springer.com/content/pdf/10.1007/s00354-019-00070-2.pdf.

[35] Nikhita Vedula, Pranav Maneriker, and Srinivasan Parthasarathy. "BOLT-K: Bootstrapping Ontology Learning via Transfer of Knowledge". In: *The World Wide Web Conference*. 2019, pp. 1897–1908. URL: https://drive.google.com/file/d/1zWe5Wyi3MQ8fV-VOry6eoiBMtgybybQs/view.

[36] Lalit Mohan Sanagavarapu, Vivek Iyer, and Y Raghu Reddy. "OntoEnricher: A Deep Learning Approach for Ontology Enrichment from Unstructured Text". In: *arXiv preprint arXiv:2102.04081* (2021). URL: https://arxiv.org/pdf/2102.04081.

[37] Yuan Ren, Artemis Parvizi, Chris Mellish, et al. "Towards competency question-driven ontology authoring". In: *11th International European Semantic Web Conference*. Ed. by V Presutti, C d'Amato, F Gandon, et al. Springer, 2014, pp. 752–767. ISBN: 3319074423.

[38] Cristóvão Sousa, António Lucas Soares, Carla Pereira, et al. "Establishing Conceptual Commitments in the Development of Ontologies through Competency Questions and Conceptual Graphs". In: *On the Move to Meaningful Internet Systems: OTM 2014 Workshops*. Springer, 2014, pp. 626–635. ISBN: 3662455498.

[39] Michael Compton, Payam Barnaghi, Luis Bermudez, et al. "The SSN ontology of the W3C semantic sensor network incubator group". In: *Journal of Web Semantics* 17 (2012), pp. 25–32. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.302.4328&rep=rep1&type=pdf.

[40] oneM2M. *TS-0012-V3.7.3 Base Ontology*. Tech. rep. oneM2M, Feb. 2019. URL: https://www.onem2m.org/images/pdf/TS-0012-Base_Ontology-V3_7_3.pdf.

[41] Laura Daniele, Frank den HArtog, and Jasper Rose. *Study on Semantic Assets for Smart Appliances Interoperability*. Tech. rep. European Commission, Directorate-General of Communications Networks, Content and Technology, 2015. URL: https://docs.google.com/file/d/0B2nnxMhTMGh4WTVsSVRsb01ha3c/edit.

[42] Victor Charpenay, Maxime Lefrancois, Maria Poveda Villalon, et al. *Thing Description Ontology W3C Editor's Draft 05 May 2021*. Tech. rep. World Wide Web Consortium, May 2021. URL: https://www.w3.org/2019/wot/td#introduction.

[43] Ioan Szilagyi and Patrice Wira. "Ontologies and Semantic Web for the Internet of Things-a survey". In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2016, pp. 6949–6954.

[44] Francesco Antoniazzi and Fabio Viola. "Building the semantic Web of things through a dynamic ontology". In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 10560–10579.

[45] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, et al. "IoT-O, a core-domain IoT ontology to represent connected devices networks". In: *European Knowledge Acquisition Workshop*. Springer. 2016, pp. 561–576.

[46] Sefki Kolozali, Maria Bermudez-Edo, Daniel Puschmann, et al. "A knowledge-based approach for real-time iot data stream annotation and processing". In: *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*. IEEE. 2014, pp. 215–222.

[47] John Soldatos, Nikos Kefalakis, Manfred Hauswirth, et al. "Openiot: Open source internet-of-things in the cloud". In: *Interoperability and open-source solutions for the internet of things*. Springer, 2015, pp. 13–25.

[48] Maria Bermudez-Edo, Tarek Elsaleh, Payam Barnaghi, et al. "IoT-Lite: A Lightweight Semantic Model for the Internet of Things and Its Use with Dynamic Semantics". In: *Personal Ubiquitous Comput.* 21.3 (June 2017), pp. 475–487. ISSN: 1617-4909. DOI: 10.1007/s00779-017-1010-8. URL: https://doi-org.ezproxy.usq.edu.au/10.1007/s00779-017-1010-8.

[49] Tarek Elsaleh, Maria Bermudez-Edo, Shirin Enshaeifar, et al. "IoT-stream: a lightweight ontology for internet of things data streams". In: *2019 Global IoT Summit (GIoTS)*. IEEE. 2019, pp. 1–6.

[50] Asunción Gómez-Pérez and Mari Carmen Suárez-Figueroa. "Scenarios for Building Ontology Networks within the NeOn Methodology". In: *Proceedings of the Fifth International Conference on Knowledge Capture*. K-CAP '09. Redondo Beach, California, USA: Association for Computing Machinery, 2009, pp. 183–184. ISBN: 9781605586588. DOI: 10.1145/1597735.1597773. URL: https://doi-org.ezproxy.usq.edu.au/10.1145/1597735.1597773.

[51] Meriem Aziez, Saber Benharzallah, and Hammadi Bennoui. "An ontology based context model for the discovery of IoT services in the Internet of Things". In: *2017 International Conference on Mathematics and Information Technology (ICMIT)*. IEEE. 2017, pp. 209–213.

[52] Krzysztof Janowicz, Armin Haller, Simon J.D. Cox, et al. "SOSA: A lightweight ontology for sensors, observations, samples, and actuators". In: *Journal of Web Semantics* 56 (2019), pp. 1–10. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2018.06.003. URL: https://www.sciencedirect.com/science/article/pii/S1570826818300295.

[53] Tarek Elsaleh, Shirin Enshaeifar, Roonak Rezvani, et al. "IoT-Stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services". In: *Sensors* 20.4 (2020), p. 953.

[54] Mohamad Gharib, Paolo Giorgini, and John Mylopoulos. "Towards an ontology for privacy requirements via a systematic literature review". In: *International conference on conceptual modeling*. Springer. 2017, pp. 193–208. URL: https://www.researchgate.net/profile/Mohamad-Gharib/publication/318787316_Towards_an_Ontology_for_Privacy_Requirements_via_a_Systematic_Literature_Review/links/597ecfc8a6fdcc1a9accba79/Towards-an-Ontology-for-Privacy-Requirements-via-a-Systematic-Literature-Review.pdf.

[55] Jing Qiu, Zhihong Tian, Chunlai Du, et al. "A survey on access control in the age of internet of things". In: *IEEE Internet of Things Journal* 7.6 (2020), pp. 4682–4696.

[56] Bruno Augusti Mozzaquatro, Carlos Agostinho, Diogo Goncalves, et al. "An ontology-based cybersecurity framework for the internet of things". In: *Sensors* 18.9 (2018), p. 3053.

[57] Mayke Ferreira Arruda and Renato Freitas Bulcao-Neto. "Toward a lightweight ontology for privacy protection in IoT". In: *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*. 2019, pp. 880–888.

[58] Faiza Loukil, Chirine Ghedira-Guegan, Khouloud Boukadi, et al. "Liopy: A legal compliant ontology to preserve privacy for the internet of things". In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 2. IEEE. 2018, pp. 701–706.

[59] John W Creswell and David J Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Ed. by David C Felts. 5th ed. Thousand Oaks, California: Sage, 2018, p. 275. ISBN: 978-1-4522-2610-1.

[60] Helen Hasan and Alanah Kazlauskas. "Making sense of IS with the Cynefin framework". In: (2009). URL: https://ro.uow.edu.au/cgi/viewcontent.cgi?article=2008&context=commpapers.

[61] Cynthia F Kurtz and David J Snowden. "The new dynamics of strategy: Sense-making in a complex and complicated world". In: *IBM systems journal* 42.3 (2003), pp. 462–483. URL: https://vdc.edu.au/wp-content/uploads/2018/02/Sense-making-in-a-complex-and-complicated-world.pdf.

[62] David J Snowden and Mary E Boone. "A leader's framework for decision making". In: *Harvard business review* 85.11 (2007), p. 68. URL: https://www.systemswisdom.com/sites/default/files/Snowdon-and-Boone-A-Leader's-Framework-for-Decision-Making_0.pdf.

[63] Simon French. "Cynefin: uncertainty, small worlds and scenarios". In: *Journal of the Operational Research Society* 66.10 (2015), pp. 1635–1645. URL: https://link.springer.com/article/10.1057/jors.2015.21#citeas.

[64] Desmond Bala Bisandu. "Design science research methodology in Computer Science and Information Systems". In: *International Journal of Information Technology* 5.4 (2016), pp. 55–60. URL: https://www.researchgate.net/publication/330041672_Design_Science_Research_Methodology_in_Computer_Science_and_Information_Systems.

[65] Philipp Offermann, Olga Levina, Marten Schönherr, et al. "Outline of a design science research process". In: Jan. 2009. DOI: 10.1145/1555619.1555629. URL: https://www.researchgate.net/publication/221581320_Outline_of_a_design_science_research_process.

[66] Renee Elio, Jim Hoover, Ioanis Nikolaidis, et al. *About computing science research methodology*. 2011. URL: https://webdocs.cs.ualberta.ca/~c603/readings/research-methods.pdf.

[67] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, et al. "A design science research methodology for information systems research". In: *Journal of management information systems* 24.3 (2007), pp. 45–77. ISSN: 0742-1222.

[68] Kent O'Sullivan. "Development of a Cyber Effects Simulation Ontology for Use in Military Simulation". Dec. 2015.

[69] Oscar Corcho, Mariano Fernandez-Lopez, and Asuncion Gomez-Perez. "Ontological engineering: what are ontologies and how can we build them?" In: *Semantic web services: Theory, tools and applications*. IGI Global, 2007, pp. 44–70.

[70] Kent Beck. *Test-driven development: by example*. Addison-Wesley Professional, 2003. ISBN: 0321146530. URL: http://www.eecs.yorku.ca/course_archive/2003-04/W/3311/sectionM/case_studies/money/KentBeck_TDD_byexample.pdf.

[71] Dejing Dou, Drew McDermott, and Peishen Qi. "Ontology translation on the semantic web, ontology, semantic, cyber". In: *Journal on data semantics II*. Springer, 2005, pp. 35–57. ISBN: 3540242082.

[72] M de Boer and JP Verhoosel. "Creating and evaluating data-driven ontologies". In: *to appear* (2019). URL: https://www.researchgate.net/profile/Lea-Daling/publication/340886820_Media_Comparison_for_Instruction-based_AR_Usage_in_Collaborative_Assembly/links/5ea2b5d092851c87d1b105aa/Media-Comparison-for-Instruction-based-AR-Usage-in-Collaborative-Assembly.pdf#page=121.

[73] Leo Obrst, Werner Ceusters, Inderjeet Mani, et al. "The evaluation of ontologies". In: *Semantic web*. Springer, 2007, pp. 139–158. URL: https://philpapers.org/archive/OBRTEO-6.pdf.