FEDERAL UNIVERSITY OF THE STATE OF RIO DE JANEIRO
CENTER OF EXACT SCIENCES AND TECHNOLOGY
SCHOOL OF APPLIED INFORMATICS

MyMemory: An ontology for privacy in external digital memories

Bianca Rodrigues Teixeira

Supervisors
Flavia Maria Santoro
Fernanda Araujo Baião

RIO DE JANEIRO, RJ – BRAZIL
DECEMBER OF 2017

Catalogação informatizada pela autora

T266 Teixeira, Bianca Rodrigues
MyMemory: An ontology for privacy in external digital memories / Bianca Rodrigues Teixeira. -- Rio de Janeiro, 2017.
78 p

Orientadora: Flavia Maria Santoro.
Coorientadora: Fernanda Araujo Baião.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2017.

1. Ontologia. 2. Memória. 3. Privacidade. I. Santoro, Flavia Maria, orient. II. Baião, Fernanda Araujo, coorient. III. Título.

MyMemory: An ontology for privacy in external digital memories

Bianca Rodrigues Teixeira

Undergraduate thesis and graduation project presented at the School of Applied Informatics at Federal University of the State of Rio de Janeiro (UNIRIO) in order to obtain the title of Bachelor in Information Systems.

Approved by:

Flavia Maria Santoro (UNIRIO)

Fernanda Araujo Baião (UNIRIO)

Kate Cerqueira Revoredo (UNIRIO)

RIO DE JANEIRO, RJ – BRAZIL
DECEMBER OF 2017

Acknowledgments

Earlier this year, my supervisor Flavia told me about this conference that was holding a workshop about Black Mirror. I knew I wanted to work with Black Mirror for my thesis, so when we heard of this workshop we figured we'd try to submit something. Even if it wasn't accepted, we could use the paper for this work.

Oddly enough, it was accepted. I'd never written a paper before, so I was quite excited. An international conference! In Vienna! It felt like a dream.

ISWC was incredible. I wasn't familiar with the world of Semantic Web, and honestly I'm still not, but the conference showed me how big the world can be. It also made me want to pursue a Master's degree, which I wasn't sure about before.

Overall, I'm extremely thankful for Flavia having pushed me into doing something I couldn't do alone. I'd like to thank Fernanda and Kate for being my conference-buddies, they made the experience so much more fun. Fernanda ended up supervising this work along with Flavia, so I'd like to apologize to them both for my annoying emails on Sunday mornings, afternoons and evenings. Thank you so much for the non-stop support and for your valuable contributions to this work. I truly appreciate it.

I'd like to thank my family and apologize for being a little off over these past weeks. I'll be better, I promise.

Also, I'd like to thank my boyfriend, Hugo, for always being amazing. I love you.

To everyone whom I met at UNIRIO, it was a pleasure. My time here has been great. I'd like to specifically thank Davi for doing me a huge favor once.

I believe this is it. I hope you'll enjoy reading this work as much as I enjoyed writing it!

Bianca

ABSTRACT

Privacy is a concerning issue today, with millions of people sharing their everyday lives on various social media platforms. Everything people share can be considered as part of their digital memory, which can be consisted of thoughts and feelings posted online. Memory is the focus of “The entire history of you”, third episode of British television series “Black Mirror”. The characters use a device that records and saves what their eyes see, and it is also possible to browse through all previous memories. However, it can be a huge problem if somebody tries to see memories that are supposed to be private. To try to combat that possibility, we defined MyMemory, an ontology which helps define the current situation the user is in, in case somebody else tries to access their memory, and automatically determines whether a certain person can or cannot access a specific user’s memory. The proposed MyMemory ontology is an extension of an existing ontology for external digital memory. We used UML to specify and OWL and Protégé to codify our ontology. We believe that there is still room for improvement and for more discussion about video-memory and privacy, which is a topic that is not frequently debated.

Keywords: ontology, memory, privacy, black mirror

RESUMO

Privacidade é uma questão preocupante hoje em dia, pois milhões de pessoas compartilham suas vidas todos os dias em diversas redes sociais. Tudo que é compartilhado por pessoas pode ser considerado parte de sua memória digital, que pode ser composta de pensamentos e sentimentos postados online. Memória é o foco de "Toda a sua história", terceiro episódio da série de TV britânica "Black Mirror". Os personagens usam um dispositivo que grava e salva tudo que seus olhos veem, e é possível navegar pelas memórias antigas. No entanto, pode surgir um grande problema no momento que alguém tenta ver memórias que deveriam ser privadas. Para combater essa possibilidade, definimos a ontologia MyMemory, que ajuda a definir a situação atual em que o usuário está, no caso de haver outra pessoa que deseja acessar sua memória, e determina se a pessoa deve ou não acessar a memória do usuário. A ontologia MyMemory proposta é uma extensão de uma ontologia existente para memória digital externa. Nós usamos UML para especificar e OWL e Protégé para codificar nossa ontologia. Nós acreditamos que ainda há espaço para melhorias e para mais discussão sobre memórias no formato de vídeo e privacidade, que não é um tópico debatido com frequência.

Palavras-chave: ontologia, memória, privacidade, black mirror.

Index

Table Index

Image Index

1 Introduction

1.1 Motivation

In the past few years, technology has experienced some major advancements. British television series “Black Mirror”[1] highlights several scenarios in which those said advancements could potentially lead to a dystopian society. As its creator Charlie Brooker stated [2], its episodes are about today’s society’s future if people are not careful. This work is concerned with the topic presented in the third episode of Black Mirror’s first season, entitled “The entire history of you”[2].

This episode is set in a world in which most humans have the “grain” device implanted in their bodies. The grain captures the images seen from the user’s eyes and saves them so that they can watch previous times of their lives. The memory can be played back only in the user’s vision or it is possible to display it on a television screen, pointing the remote that comes with the grain to the screen. It acts as a digital memory, browsable and rewindable. That device makes sense (and may even alert for the possible consequences of it) since, every day, over one billion people actively use Facebook[3] and other social media, sharing their opinions, describing facts they believe, and detailing moments of their lives online [4]; and all this data has been stored and can be revisited on demand. In some way, we are already digitizing part of our lives and, thus, contributing to creating external digital memory.

However, “The entire history of you” leaves a gap on a current and very relevant issue, which is privacy. Acquisiti and Gross [8] performed a study and found that a

[1] http://www.channel4.com/programmes/black-mirror/
[2] http://www.imdb.com/title/tt2089050/
[3] http://facebook.com

minimal percentage of Facebook users even change the default privacy settings. Moreover, this issue has been addressed through the concept of Privacy-by-Design which is related to a proactive incorporation of privacy principles in a system's design, i.e. an approach to minimize information systems' privacy risks through technical and governance rules [21].

In this work, we explore the issue of privacy, inspired by the memory recording technology presented in "The entire history of you". The episode's main character, Liam, suspects his wife Ffion has romantic feelings toward Jonas, an old friend of hers whom he met at a dinner party. There, Jonas mentioned his habit of watching old memories with previous lovers, which made other guests feel uncomfortable. Later that day, Liam uses his own memory to analyze Ffion's behavior when she was with Jonas and confronts her about it. As it turned out, Ffion and Jonas were indeed in a relationship in the past, which made Liam feel anger and jealousy toward Jonas and his habit of revisiting old memories. Liam, drunk, went to Jonas' house and violently forced him to erase every video of Ffion stored in his Grain and made him show it to him on the television. Later, Liam noticed, after watching the memory of Jonas erasing said videos, that one particular memory was set in his own home, meaning it happened during his marriage with Ffion. Because of the date of that encounter, Liam started to wonder if Ffion's daughter was actually his or Jonas'. He, then, made Ffion show him, on the TV, her memory of that day, to check if Jonas used a condom, which we believe violated her privacy. Liam forced himself into seeing other people's memories, threatening their safety. So, the privacy aspect of memory needs to be addressed in order to avoid such situations. We argue that the issue of privacy concerning digital memory illustrated in this science fiction story is an important discussion that should be made within the Information Systems area and represents the main focus of this undergraduate work.

1.2 Objectives

Based on the scenario described and the work of [19], we propose to address the privacy issue of external digital memories by applying conceptual modelling and web semantics technologies. For that, we specified, codified and illustrated the use of an ontology for the broad concept of digital memory that includes the concept of private memory associated to the context in which it was produced. We named it MyMemory. Regarding “The entire history of you”, we aim at specifying a technique for the grain device to be able to customize and automatically control unwanted requests from third parties to view a person's memories in certain situations of privacy violation.

The ontology was built as an extension of the work described in [19]. We specified the ontology as a UML conceptual diagram, and then used the software Protégé and OWL to codify our ontology. The main use of our OWL ontology is to automatically detect situations in which the owner of a certain memory is being requested to share that memory but does not want to share it. Our ontology encompasses logic axioms to support inferring whether the person that wants to see the owner’s memory should have access or not to that memory. Therefore, we rely on inferences to protect the user’s privacy.

1.3 Structure

This work is structured in sections. Apart from the introduction, it features the following sections:

- Section 2, or Important Concepts, features an overview of topics we referred to a lot in this work. It presents an introduction on memory and its relationship with privacy, introduces the concept of Ontology, mentioning some characteristics of ontology languages, specifically OWL, which we use on this work, and we present the software Protégé and the features we used for this work.

- Section 3, or MyMemory, is the formal definition of the ontology we developed. We analyse the original External Digital Memory Ontology, which is the base for our ontology, and present our ontology and its characteristics, getting into detail about its classes, its object properties, its data properties and its rules.

- Section 4, or Using MyMemory, has two examples of scenarios in which our ontology is used. The first is an example of a situation in which a memory should be private and the second is an example of a memory being public. We also discuss some limitations of our ontology and then its benefits.

- Finally, Section 5, or Conclusions, closes off this work. We give our final remarks, discuss more limitations of what we presented and provide ideas for future work.

2 Important Concepts

This section presents important concepts this work is based on.

Section 2.1 presents an overview of the concept of Memory as studied by historians and psychologists. We also briefly relate memory and privacy as seen on the Black Mirror episode.

Section 2.2 features a discussion on ontology, which is the technology we are using to address the privacy issue the episode portrayed, along with the language and tool we used.

2.1 Memory

Le Goff, french historian, claims that the concept of memory relates to an individual and psychological event which allows for humans to update past impressions or information they previously had [14]. According to Gondar [6], memory would be characterized as a personal and subjective experience, lacking documentation. Because of that, the sharing of individual memory is possible through speech.

According to Megill [16], "*Memory is an image of the past constructed by a subjectivity in the present*." The operations related to memory – the acts of remembering and forgetting – undergo an ontological shift with a possibility of also representing and recording both thoughts and speech established within social groups as an extension of itself, through the modeling, construction, and organization of external memories [19].

Information technology artifacts derive from an infrastructure capable of storing, generating and manipulating all the online traces created by individuals and by society. This organization includes sophisticated computational models and algorithms of semantical data processing, which are property of public or private institutions able to conduct social engineering processes in all spheres: political, civic, commercial, and individual. Individuals are the providers of their own memories, which are collected and manipulated by the real holders of power.

2.1.1 Privacy and Memory

According to Daniel Solove, an expert in privacy law, for ages, people have debated issues of privacy, ranging from gossip to surveillance [20].

> *"Privacy, however, is a concept in disarray. Nobody can articulate what it means. Currently, privacy is a sweeping concept, encompassing (among other things) freedom of thought, control over one's body, solitude in one's home, control over personal information, freedom from surveillance, protection of one's reputation, and protection from searches and interrogations." Solove, D.* [20]

In the context of "The entire history of you", the grain's users record basic everyday tasks as well as their most intimate moments. The "solitude in one's home", as mentioned above, is recorded through one's eyes.

The idea of the grain technology is that only the user has access to his/her memories, since it's a personalized device and is accessed through his/her own eyes. However, because of the possibility to play back the video on a television screen, the grain creators stimulate people to share their memories with other people, instead of it being possible only to watch through the user's personal vision.

In order for a user to share a specific memory with somebody else, the user needs to use the remote that comes with the grain and find that memory. According to the episode, that is something only the owner of the memory can do. This creates a feeling of privacy: the memories are stored on the user's grain and only the user has access to it, regardless of the content of the memory. If it was recorded by the user's eyes, the user controls it.

However, as it can also be seen on the episode, it's easy to *make* the user show his or her memories to somebody else. If the user is being threatened, it leaves no other choice other than to access the grain and show or delete the memories, as per somebody else's request, or on demand. Therefore, the user's alleged "guaranteed" privacy is broken. The grain technology makes possible for it's users' most intimate, private moments to be violated.

In this work, we propose a way to avoid scenarios like these. So, as a starting point, we argue for a formal definition of memory and privacy through an ontology extended from [19] and [15].

2.2 Ontology

An ontology is a formal, explicit specification of a shared conceptualization [22]. The term originated in philosophical studies, in which it is considered a systematic account of Existence [9].

An ontology in Computer Science or Information Systems highlights relevant entities and the relationships between them. The backbone of an ontology, according to Guarino et al. (2009), consists of a generalization/specialization of concepts [11].

> *"For appropriate usage, ontologies need to fulfill a further function, namely facilitating the communication between the human and the machine – referring to terminology specified in the ontology – or even for facilitating inter-machine and inter-human communication. " Guarino et al.* [11]

In the context of Semantic Web, as defined by [7], ontologies "describe domain theories for the explicit representation of the semantics of the data", providing formal conceptualizations. The final goal of an ontology is to specify a computational artifact, so that it can become a model, or a reference, in its domain.

Guarino defines four types of ontology, as described below: [10]

- *Top-level ontologies* describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain: it seems therefore reasonable, at least in theory, to have unified top-level ontologies for large communities of users.
- *Domain ontologies* describe the vocabulary related to a generic domain (like medicine, or automobiles).
- *Task ontologies* describe a generic task or activity (like diagnosing or selling), by specializing the terms introduced in the top-level ontology.
- *Application ontologies* describe concepts depending both on a particular domain and task, which are often specializations of both the related ontologies.

An ontology is composed of several elements, such as classes, object properties (relationships between classes), data properties (or attributes), axioms (which define logical rules that generate inferences) and individuals (instances of classes).

This work's objective is to build an ontology on Memory, which will be specifically used to determine the privacy of a memory being requested. So, we are presenting a domain ontology.

The concept of Ontology Engineering helps develop an ontology: the first step is specification, in which is created a reference model, and then there is codification, in which an ontology language is used. It is also possible to "reverse engineer" an existing ontology as seen in [12].

2.2.1 Ontology languages overview

Recently, several markup languages have been developed with the Semantic Web in mind [18]. In order to create a new language, it must meet the following seven requirements [5]:

- Have a compact syntax

- Be highly intuitive to humans
- Have a well-defined formal semantics
- Be able to represent human knowledge
- Include reasoning properties
- Have the potential for building knowledge bases
- Have a proper link with existing web standards to ensure interoperability

A Semantic Web language must be machine-learning readable [18], which is what we are aiming for on this work.

2.2.2 OWL

In order for the ontologies to be shared online, the Web Ontologies Language, or OWL, was developed, by the World Wide Web Consortium (W3C) Web Ontology Working Group. OWL represents information about objects and how they relate to each other and also represents information about the object itself, which can simply be called "data" [13]. It is developed as a vocabulary extension of RDF (Resource Description Framework) and it is derived from the DAML+OIL Web Ontology Language[4].

For this work, we used the software Protégé, running on OWL to formally codify our ontology on Memory.

2.2.3 Protégé

Protégé is a free, open-source ontology editor and framework for building intelligent systems[5]. It was developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine. It is highly used in the academic community and it is the tool we used to build the ontology presented on this work.

It provides a visual interface for creating and defining an ontology. The ontology we are presenting on this work was made entirely using the following features available on Protégé:

---

[4] https://w3.org/TR/owl-ref/
[5] https://protege.stanford.edu

- Create class/subclass
- Create object property
    - An object property can also be defined as a relationship between two classes.
- Create data property
    - A data property can be defined as an attribute for a class.
- Create individual
    - An individual is an instance of a class.
    - For example, an instance of the class Dog could be Toto, an instance of the class Person could be Dorothy and Dorothy could have the object property *isOwnerOf Toto*.
- Class expressions
    - A class expression relates a class to another class using an object property and a keyword that represents restrictions.
    - For example, the class expression "hasPet some Dog" means "Thing that has a pet that is a Dog". Individuals that are instances of this class expression have a relationship along the hasPet property to an individual that is an instance of class Dog. [17]
    - Another example would be "hasPet exactly 2 GoldFish". It means "Things that have exactly 2 GoldFish as pets". A class that has this expression can have other pets, but two of them must be of the type GoldFish [17].
- Create rule
    - Rules in Protégé are written in the Semantic Web Rule Language and are expressed using OWL concepts (classes, properties, individuals).
    - Rules are powerful as they can reclassify an individual (change its type), they can assign a value to a property and more.
    - Here is an example: Person(?p), Dog(?d), isOwnerOf(?p, ?d) → DogOwner(?p).
    - The rule above states that, if a person ?p owns a dog, then ?p extends from the type DogOwner (reclassification of individual). ?p was

previously classified as a Person, but, because of the rule, he/she is also classified as a DogOwner.

- Start reasoner/Synchronize reasoner
    - Reasoner is a software that is used to derive new facts from the existing ontologies [1]. It "infers logical consequences from a set of explicitly asserted facts or axioms and typically provides automated support for classification, debugging and querying" [3]. Also, it detects inconsistencies defined in the ontology.
    - Using the rules we have defined on our ontology, *inference* plays a big part on what we are proposing, because of the automatization of reclassifications and property values assignments. Once the reasoner synchronization button is clicked on Protégé, it gathers all the manually defined properties and classifications and, based on the rules that have been previously defined, checks if any new classifications or properties should be inferred. It checks every individual to see if any of its properties or classifications fall into any of the rules. If it does, it infers new characteristics for the individual.
    - The reasoner we are using is HermiT[6], a default on Protégé. It can be synchronized in seconds, making it easy to check the ontology. According to its web page, it can, given an OWL file, "determine whether or not the ontology is consistent, identify subsumption relationships between classes, and much more". It plays a big part in our ontology, because the inferences are what ultimately determine whether a memory should be accessed or not.

Using the features we just described, we came up with our ontology.

The following section features a more elaborate discussion about the ontology on Memory we have developed.

[6] http://www.hermit-reasoner.com

3 MyMemory

Section 3.1 presents the External Digital Memory Ontology, which is an existing ontology this work is based on. However, it was not formally defined in OWL, only in UML, and did not fully address the privacy issue, which is the focus of this work. We reused the classes and properties defined, added more classes and properties and were able to generate a code for our ontology in OWL.
Section 3.2 presents the entities of our ontology, along with their relationships and also some rules we created to be able to infer on new characteristics. An earlier version of this ontology is described in [23]. The goal of our new ontology is to protect a user's memories from being watched by someone else in a situation in which the user does not want to share his/her memories. The original idea is that it could be used by the grain technology depicted in Black Mirror, but any information system that deals with memory could reuse it if it intends on protecting the users's privacy.

3.1 External Digital Memory Ontology

Rodrigues et al. [19] state that thinking memory opens the possibility that, from a new situation or a new encounter, the past can be both remembered and reinvented. In this way, the history of a subject, individual or collective, can be the history of the different senses that emerge in their relationships. It makes possible that memory, instead of being recovered or rescued, can be created and recreated from the new senses that always occur for both individual and collective memory, since they are all social subjects. The polysemy of memory, which could be its flaw, is indeed its richness. Rodrigues et al. [19] proposed a preliminary ontological model to represent the concept of external digital contemporary memory which encompasses classes (concepts) and their relationships (Figure 1). It is called External Digital Memory Ontology.

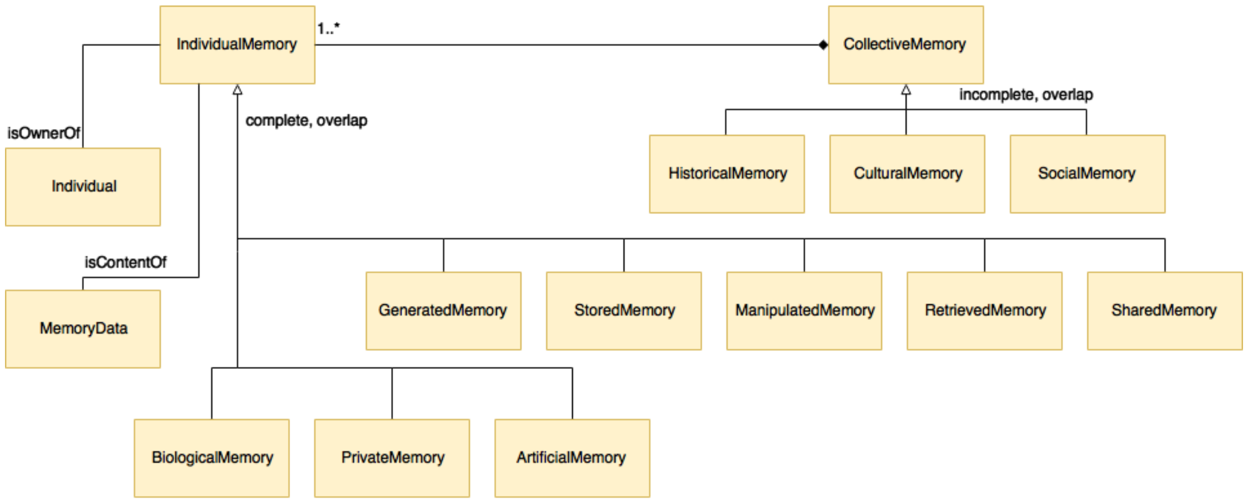

Figure 1 - Entities and relationships of the External Digital Memory Ontology in UML

Collective memories result from the interactions between individuals, the past and present reality. The processes of collective memory production are diverse and transformed over time according to the social experience and the technological devices of that context. Fundamental operations, such as, generate, store, manipulate, modify, retrieve, and share, affect external memory. So, the creation of digital memories occurs through various processes, through social networks on the web, applications, surveillance cameras in public and private space, closed systems, content production, GPS, cookies, digital traces, among many others. The production of the memory operates in "real-time" or "online" both in individual and in collective layer. Thus, digital memory can provide relationships between individual and collective memories, and might subvert ontological experiences, such as the processes of remembering and forgetting, as well as cultural, social, economic, political, historical, and other issues.

Table 1 explains the types of memory that compose the main concept of external digital memory.

Memory Type	Content
Individual	Part of the external memory that exchanges information shared with the external world to the cyberspace.
Collective	Produced by relationships and experiences practiced among individuals and social groups through interactions with individual memory and auto- relations
Biological	Born with the individual
Private	Proprietary, not shared by any individual, organism or thing
Shared	Proprietary that interacts with the cyberspace; Distributed by an individual, collective or system agent
Artificial	Any non-biological memory of the individual
Generated	Generated or created by individuals or collectivity, explicitly or implicitly
Stored	Registered and stored within private, commercial, or governmental clouds
Manipulated	Operated by applications, social networks or systems that access the cyberspace
Retrieved	Accessed for private or collective use, or yet to obtain digital traces
Modified	Subjected to changes, with/without authorization of the individual or collective agent

Table 1 - Types of memory defined in External Digital Memory Ontology

This ontology had not been implemented yet and had no actual application potential in Information Systems. This work reuses it and provides a use case for it.

3.2 MyMemory

Analyzing the dystopian future of Black Mirror, we first observe that people's memories are reduced to videos of interactions of the human eye with certain events, like a camera that can only capture a collection of instant pictures. Our ontology intends to broaden this concept. We are interested in securing individual memories, identifying the context of one particular piece of memory, and moreover, in establishing boundaries able to distinguish what should be public and what should be private [23]. In this sense, we extended the ontology presented in [19] to include several new classes and relationships to help secure somebody’s memories based on a current Situation. Appendix A contains the full OWL code for our ontology. Also, since the grain is a personal system, we focused on the concept of Individual Memory and on protecting its privacy. Figure 2 presents the graphical representation for the ontology organized in a Unified Modeling Language (UML) Class Diagram.

For visualization purposes, we have decided to illustrate only IndividualMemory and Person from the original External Digital Memory Ontology. Note that the original External Digital Memory Ontology used the term "Individual" to identify a person, but this term is a reserved word for ontologies, since it represents an instance of a class. Therefore, we renamed it Person.

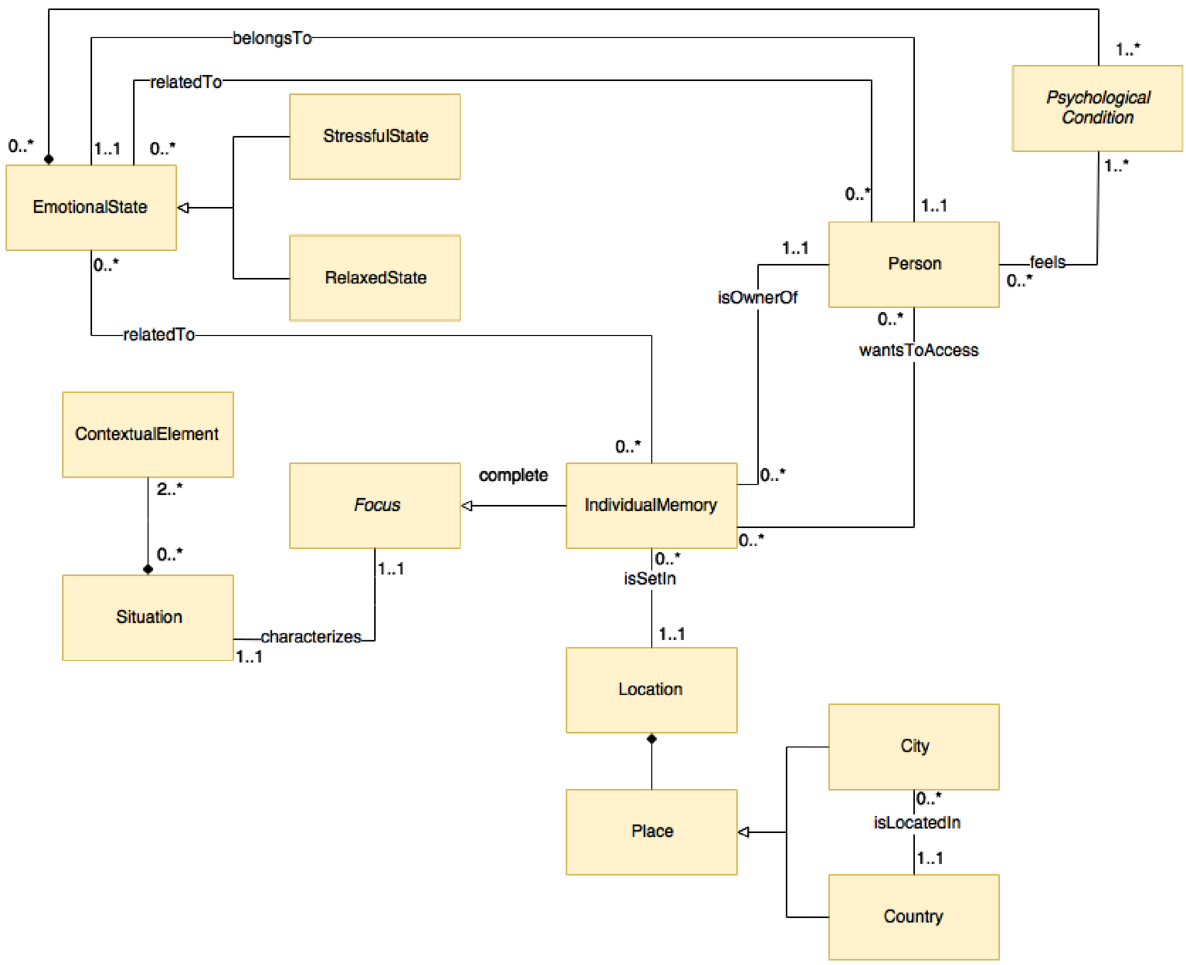

Figure 2 - Visualization of the entities of MyMemory (incomplete for visualization purposes)

It is of great importance to note that, for visualization purposes, we did not include subclasses of PsychlogicalCondition. **Happy**, **Nervous** and **Hesitant** are subclasses of PsychologicalCondition manually defined in our ontology, but the idea is that every possible feeling falls under PsychologicalCondition, as a subclass.

PsychologicalCondition is merely an abstract class. Therefore, an EmotionalState is a composition of various subclasses of PsychologicalCondition.

Also, an EmotionalState can be classified as a RelaxedState or a StressfulState, based on the PsychologicalConditions it has. We have defined these two states, but there can be more states with other combinations of PsychologicalConditions. There can be multiple definitions of the type of EmotionalState that is ongoing, which makes it not deterministic. The subclasses of EmotionalState help identify how the owner of the memory is feeling, which helps define whether a particular memory should be shared or not. So, although we have defined only Relaxed and Stressful states, there can be more.

We consider Focus what is currently under analysis, which is an IndividualMemory. It is an abstract class, whose purpose is to give an IndividualMemory a different semantic other than someone's memory. A Situation characterizes a Focus, which relates to a memory since the point of this ontology is to determine the privacy of a certain memory. However, the piece of memory is always the same, what changes is the Focus and current Situation. Also, a Situation is composed of ContextualElements, which can be small, individual details about what is happening at that time. A combination of ContextualElements define a Situation, which characterizes a Focus, and ultimately, characterizes an IndividualMemory. Our rules state three possible ContextualElements, as presented in Section 3.2.4.

3.2.1 Classes

Table 2 defines each class present in the visualization depicted in Figure 2.

Class	Meaning
IndividualMemory	Piece of memory stored on a user's grain
Person	Could be any person, e.g. the owner of the memory or someone that wants access to that particular memory.
Focus	What is currently under analysis, i.e. an IndividualMemory.
ContextualElement	Is something that composes a Situation.
Situation	Characterizes a Focus (or a memory) and is composed of ContextualElements.
EmotionalState	Is a composition of PsychologicalCondition and represents how a user is feeling toward someone else and their desire to access the user's memory.
PsychologicalCondition	Is an abstract class and is the superclass of other feelings.
StressfulState	Is an EmotionalState defined as stressful based on the PsychologicalConditions it has.
RelaxedState	Is an EmotionalState defined as relaxed based on the PsychologicalConditions it has.
Location	An IndividualMemory is set in a *Location*, which is composed of different Places (a City and a Country, for example).
Place	Superclass to different possible places.
Country	Is a Place and defines a world country.
City	Is a place and defines a city, which is located in a Country.

Table 2 - Classes from MyMemory and their definitions

3.2.2 Object Properties

Table 3 defines the object properties present in our ontology. Some properties have a "domain" and a "range". A domain is the type that has the property, and the range is the type that has the value of the property. For example, in the first row is "belongs to", whose domain is EmotionalState and whose range is Person. This means that, if that property is used, it should be a property of an individual of type EmotionalState and should have as a value an individual of the type Person. So, an EmotionalState belongs to a Person.

If a property has an inverse property, it means that the domain and range of the inverse property would be inverse to the original property. So, following the same example, "experiences" is the inverse of "belongsTo", so its domain would be Person and its range would be EmotionalState. In order words, a Person experiences an EmotionalState.

If two properties a and b are disjoint, this means that no individual x can be connected to individual y through both a and b properties at the same time.

Note that some properties do not have domain and range specified. Domain and range of properties that are inverse of other properties that do have their domain and range specified do not need to be manually defined.

There are two properties, however, "partOf" and "hasPart" (one is the inverse of the other) that do not have domain and range defined. The reason for that is that these properties represent the pre-defined and specific semantics of *composition*, in the mereology sense. Some classes in our ontology are *composed of* other classes through this relationship. So, since there are multiple classes that use the same property, with different domains and ranges, we did not specify these characteristics.

Object Property	Domain	Range	Inverse Of
belongsTo	EmotionalState	Person	experiences
characterizes	Situation	Focus	isCharacterizedBy
experiences	Person	EmotionalState	belongsTo
feels	Person	PsychologicalCondition	isFeltBy
hasPart			partOf
isCharacterizedBy			characterizes
isFeltBy			feels
isLocatedIn	City	Country	
isOwnedBy			isOwnerOf
isOwnerOf	Person	IndividualMemory	isOwnedBy
isSetIn	IndividualMemory	Location	
isTryingToBeAccessedBy			wantsToAccess
partOf			hasPart
relatedTo	EmotionalState	IndividualMemory **or** Person	
wantsToAccess	Person	IndividualMemory	

Table 3 - Object properties, their domains, their ranges, their inverse properties and their disjoint properties from MyMemory

3.2.3 Data Properties

We have only one Data Property, which is "meaning". It is a property for the Location class and it is a string. It creates semantic for a certain location, e.g. user’s house, zoo, etc.

3.2.4 Rules

The ontology has thirteen rules that specifies the semantics of some classes or objectProperties more precisely . The set of rules represents the core of our ontology with regard to its intended use, that is, the characterization of customizable situations in which a memory should automatically be kept private from someone that wants to have access to it. Each rule is presented as follows, both in a textual description and in SWRL (Semantic Web Rule Language), for clearer understanding.

Rule 1 (Defining Person as a contextual element): In case a person would like to access a certain memory, then that person is a contextual element.

Rule 1 is implemented in SWRL as:

```
Person(?p), wantsToAccess(?p, ?iM), IndividualMemory(?iM) →
ContextualElement(?p)
```

Rule 2 (Defining Location as a contextual element): In case a memory is set in a certain location, then that location is a contextual element.

Rule 2 is implemented in SWRL as:

```
IndividualMemory(?iM), Location(?l), isSetIn(?iM, ?l) →
ContextualElement(?l)
```

Rule 3 (Owner can access his/her memory): In case a person would like to access a certain memory and is the owner of that memory, then the person can access the memory.

Rule 3 is implemented in SWRL as:

```
Person(?p), IndividualMemory(?m), wantsToAccess(?p, ?m),
isOwnerOf(?p, ?m) → canAccess(?p, ?m)
```

Rule 4 (Defining Person as not owner of a memory): In case a person p1 would like to access a memory m1, a person p2 owns m1 and p1 and p2 are different people, then p1 is not the owner of m1 (since a memory can have only one owner).

Rule 4 is implemented in SWRL as:

```
Person(?owner), IndividualMemory(?iM), Person(?requester),
wantsToAccess(?requester, ?iM), isOwnerOf(?owner, ?iM),
DifferentFrom (?owner, ?requester) →
isNotOwnerOf(?requester, ?iM)
```

Rule 5 (Defining Person as part of a Situation): In case a situation characterizes a memory and that memory is trying to be accessed by a person, than that person, as a contextual element, should be part of the same situation.

Rule 5 is implemented in SWRL as:

```
Situation(?s), characterizes(?s, ?iM),
IndividualMemory(?iM), Person(?p), wantsToAccess(?p, ?iM) →
hasPart(?s, ?p)
```

Rule 6 (Defining Location as part of a Situation): In case a situation characterizes a memory and that memory is set in a particular location, then that location should be part of the situation (as a contextual element).

Rule 6 is implemented in SWRL as:

```
Situation(?s), characterizes(?s, ?iM),
IndividualMemory(?iM), Location(?l), isSetIn(?iM, ?l) →
hasPart(?s, ?l)
```

Rule 7 (Defining EmotionalState as part of a Situation): In case a situation characterizes a memory and that memory is related to an emotional state (see rule below), then the same situation contains the emotional state as a contextual element.

The emotional state of the owner of the memory is directly related to the memory that is being requested. If the owner doesn't want to show his/her memory, the emotional state will be a stressful or an angry state, and if the owner has no problem in sharing his/her memory, he/she will feel relaxed.

Rule 7 is implemented in SWRL as:

```
Situation(?s), characterizes(?s, ?iM),
IndividualMemory(?iM), relatedTo(?eS, ?iM),
EmotionalState(?eS) → hasPart(?s, ?eS)
```

Rule 8 (Defining an EmotionalState as a contextual element, defining its feelings and defining its relationship with a person and with a memory): In case a person feels a psychological condition and he/she is the owner of a memory that somebody else is trying to access, and if the owner of the memory has an emotional state, then this means the emotional state is affected by the fact that someone else wants access to his/her memory. So, the emotional state is reclassified as a contextual element and it contains the psychological condition the user is feeling at that time. Also, the emotional state is related to the person that wants access to the owner's memory and is related to the memory itself.

Rule 8 is implemented in SWRL as:

```
Person(?p), PsychologicalCondition(?pC), feels(?p, ?pC),
EmotionalState(?eS), IndividualMemory(?iM), isOwnerOf(?p,
?iM), Person(?p2), wantsToAccess(?p2, ?iM), belongsTo(?eS,
?p) → ContextualElement(?eS), hasPart(?eS, ?pC),
relatedTo(?eS, ?p2), relatedTo(?eS, ?iM)
```

Rule 9 (Classifying EmotionalState as stressful): In case an emotional state contains the psychological condition Nervous, then it is reclassified as a stressful state.

Rule 9 is implemented in SWRL as:

```
EmotionalState(?eS), Nervous(?n), hasPart(?eS, ?n) →
StressfulState(?eS)
```

Rule 10 (Classifying EmotionalState as stressful 2): In case an emotional state contains the psychological condition Hesitant, then it is reclassified as a stressful state.

Rule 10 is implemented in SWRL as:

```
EmotionalState(?eS), Hesitant(?h), hasPart(?eS, ?h) →
StressfulState(?eS)
```

Rule 11 (Classifying EmotionalState as relaxed): In case an emotional state contains the psychological condition Happy, then it is reclassified as a relaxed state.

Rule 11 is implemented in SWRL as:

```
EmotionalState(?eS), Happy(?h), hasPart(?eS, ?h) →
RelaxedState(?eS)
```

Rule 12 (Memory can be accessed if owner feels relaxed): There is a situation that currently characterizes the memory that is trying to be accessed by a person. If the situation is composed of a relaxed emotional state, meaning, if the owner feels relaxed about someone trying to access their memory, then the person that wants access **can** access the memory.

Rule 12 is implemented in SWRL as:

```
IndividualMemory(?iM), Situation(?s), RelaxedState(?rS),
hasPart(?s, ?rS), characterizes(?s, ?iM), Person(?p),
wantsToAccess(?p, ?iM), relatedTo(?rS, ?p) → canAccess(?p,
?iM)
```

Rule 13 (Memory cannot be accessed if owner feels stressed and if someone else is requesting it): There is a situation that currently characterizes the memory that is trying to be accessed by a person that is not the owner of the memory. If the situation is composed of a stressful emotional state, meaning, if the owner feels stressed about someone else trying to access their memory, then the person that wants access **cannot** access the memory.

Rule 13 is implemented in SWRL as:

```
IndividualMemory(?iM), isNotOwnerOf(?p, ?iM),
wantsToAccess(?p, ?iM), relatedTo(?sS, ?p), hasPart(?s,
?sS), Person(?p), characterizes(?s, ?iM), Situation(?s),
StressfulState(?sS) → cannotAccess(?p, ?iM)
```

MyMemory is composed of the classes, properties and rules mentioned above. The following section features two use cases in which our ontology will determine the capacity of a person to access someone else’s memory.

4 Using MyMemory

In this section, we present a preliminary evaluation of the application of the ontology proposed within scenarios inspired by "The entire history of you". We provide a few examples of possible instantiations of the ontology using Individuals, that is, simulating scenarios of possible worlds.

4.1 Scenario 1 - Private memory

The first example is a situation from "The entire history of you".

4.1.1 Scenario description

Liam wants Ffion to show him a memory of hers with Jonas, an affair she had during their marriage. Ffion, as seen on the episode, even tells Liam she deleted the memory, because she didn't want to show it to him. In that sense, we believe Liam shouldn't have access to the memory, because Ffion, the owner, doesn't want that to happen. We believe that the owner's privacy should be respected if they don't want to share their memories, which is their right. Liam pressured her into showing him and she did, for she believed she had no other choice.

4.1.2 Ontology instantiation

To characterize the situation, we have created ten individuals in Protégé, as seen on Table 4.

Individual Name	Type
Ffion	Person
Liam	Person
FfionsMemoryWithJonas	IndividualMemory
EmotionalState1	EmotionalState
Hesitant	PsychologicalCondition
Nervous	PsychologicalCondition
Location1	Location
Bristol	City
England	Country
Situation1	Situation

Table 4 - Individuals and their types for the first scenario using MyMemory

Figure 3 is a screenshot taken from Protégé regarding the individuals mentioned above. Each individual was created clicking on the top left icon and a type, or a class, was specified for each individual according to Table 4.

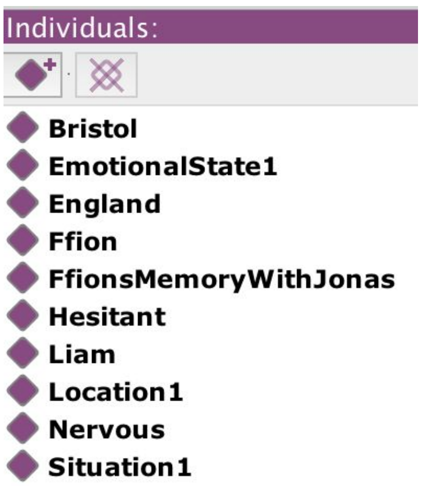

Figure 3 - Protégé screenshot of individuals for first scenario using MyMemory

Each individual from Table 4 has their own set of object properties. Table 5 illustrates the object properties of individuals and the individuals they relate to. Note that some individuals do not have any object properties defined.

Object property
feels(Ffion, Nervous)
feels(Ffion, Hesitant)
isOwnerOf(Ffion, FfionsMemoryWithJonas)
experiences(Ffion, EmotionalState1)
wantsToAccess(Liam, FfionsMemoryWithJonas)
isSetIn(FfionsMemoryWithJonas, Location1)
hasPart(Location1, Bristol)
isLocatedIn(Bristol, England)
characterizes(Situation1, FfionsMemoryWithJonas)

Table 5 - Object properties for the first scenario using MyMemory

Situation1 defines the current situation in which FfionsMemoryWithJonas is required. This way, it is composed of some contextual elements, which are: how the owner is feeling, who is requesting the memory, and where the memory was 'recorded'. However, these properties are not defined manually, they will be inferred based on what was defined in Table 5.

Figure 4 is another screenshot from Protégé, focusing on FfionsMemoryWithJonas and the details described in Table 4 and Table 5.

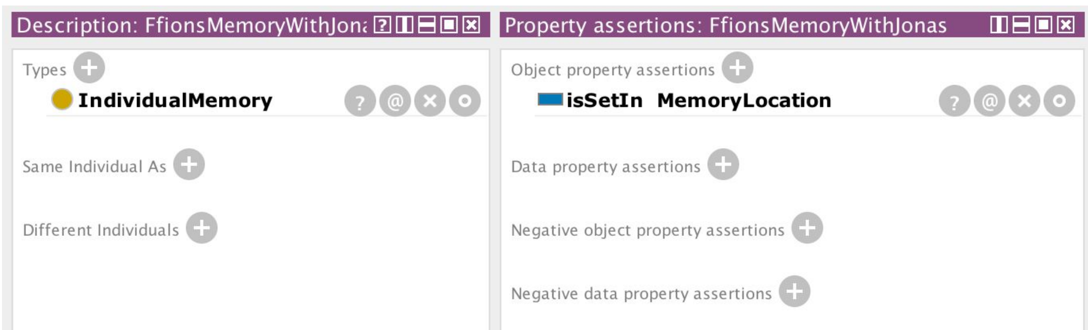

Figure 4 - Protégé screenshot showing the type of FfionsMemoryWithJonas and its object property for the first scenario using MyMemory

As it is possible to observe in Figure 4, FfionsMemoryWithJonas is an individual of type IndividualMemory and has the property isSetIn Location1.

The object properties defined in Table 5 were set to its respective individuals using the plus sign next to "Object property assertions", as seen on Figure 4.

4.1.3 Ontology use

The idea is that the grain would be able to detect the individuals from Table 4 and their object properties from Table 5 instantly. It should be able to understand that Ffion feels nervous and hesitant and that Liam wants to see that particular memory of her encounter with Jonas. Our ontology is helpful to determine what should be identified by the grain. The grain records video and audio and follows the user’s everyday life, so it could have some understanding about him/her (his/her habits, friends, etc.).

What we have presented so far in this scenario is what primarily is done by the grain. However, based on our rules and on our ontology, many other properties and concepts could be inferred, and they help determine whether Liam should be able or not to access Ffion’s memory. Therefore, the implementation of the ontology should support making inferences about the user memories.

In Protégé, once we synchronize the reasoner, it shows all the inferences that have been calculated. Table 6 shows all the new Object Properties that have been inferred.

Inferred Object Property
feltBy(Nervous, Ffion)
partOf(Nervous, EmotionalState1)
feltBy(Hesitant, Ffion)
partOf(Hesitant, EmotionalState1)
hasPart(EmotionalState1, Nervous)
hasPart(EmotionalState1, Hesitant)
belongsTo(EmotionalState1, Ffion)
relatedTo(EmotionalState1, Liam)
relatedTo(EmotionalState1, FfionsMemoryWithJonas)
partOf(EmotionalState1, Situation1)
partOf(Bristol, Location1)
partOf(Location1, Situation1)
hasPart(Situation1, Location1)
hasPart(Situation1, EmotionalState1)
hasPart(Situation1, Liam)
isOwnedBy(FfionsMemoryWithJonas, Ffion)
isCharacterizedBy(FfionsMemoryWithJonas, Situation1)
isTryingToBeAccessedBy(FfionsMemoryWithJonas, Liam)
cannotBeAccessedBy(FfionsMemoryWithJonas, Liam)
partOf(Liam, Situation1)
isNotOwnerOf(Liam, FfionsMemoryWithJonas)
cannotAccess(Liam, FfionsMemoryWithJonas)

Table 6 - Inferred object properties for the first scenario using MyMemory

Some object properties have been inferred based on the **inverseOf** property. "feltBy" was inferred because of the "feel" property, since they are inverse. Similarly, "isTryingToBeAccessedBy" is the inverse of "wantsToAccess", which was defined manually, as "isOwnedBy", which is the inverse of "isOwnerOf". Also, "isCharacterizedBy" is the inverse of "characterizes", as well as "belongsTo(EmotionalState1, Ffion)" and "experiences(Ffion, EmotionalState1)". The property "hasPart(Location1, Bristol)" has been manually added, so "partOf(Bristol, Location1)" was inferred. Other uses of "partOf" and "hasPart" in Table 6 represent inferences, since neither was manually added.

"hasPart(EmotionalState1, Nervous)", "hasPart(EmotionalState1, Hesitant)", "relatedTo(EmotionalState1, Liam)" and "relatedTo(EmotionalState1, FfionsMemoryWithJonas)" have been inferred based on the same rule: **Rule 8**. Rule 8 states that *"In case a person feels a psychological condition and he/she is the* *owner of a memory that somebody else is trying to access, and if the owner of the memory has an emotional state, then this means the emotional state is affected by the fact that someone else wants access to his/her memory. So, the emotional state is reclassified as a contextual element and it contains the psychological condition the user is feeling at that time. Also, the emotional state is related to the person that wants access to the owner's memory and is related to the memory itself."* Breaking it down in parts, we have Ffion, who feels Nervous/Hesitant. Also, we have the property "isOwnerOf(Ffion, FfionsMemoryWithJonas)". We have Jonas, who "wantsToAccess(Jonas, FfionsMemoryWithJonas)". The property "experiences(Ffion, EmotionalState1)" states that Ffion has an emotional state. Because of these properties, EmotionalState1 is classified as a ContextualElement and the property "hasPart(EmotionalState1, Hesitant/Nervous)" is defined. Also, the properties "relatedTo(EmotionalState1, Jonas)" and "relatedTo(EmotionalState1, FfionsMemoryWithJonas)" were inferred. Additionally, "partOf(Nervous/Hesitant, EmotionalState1)" have been inferred because of inverseOf qualities.

The property "hasPart(Situation1, EmotionalState1)" has been inferred based on **Rule 8** and on **Rule 7**. Rule 7 states that: *"In case a situation characterizes a memory and that memory is related to an emotional state (see rule 8), then the same situation contains the emotional state as a contextual element".* The property "characterizes(Situation1, FfionsMemoryWithJonas)" was previously defined and, based on the inferences mentioned above, we have "relatedTo(EmotionalState1, FfionsMemoryWithJonas)". So, the property "hasPart(Situation1, EmotionalState1)" correlates the situation with the emotional state since they both are related to the same memory. Inversely, "partOf(EmotionalState1, Situation1)" has also been inferred.

Next, the properties "hasPart(Situation1, Location1)" and "hasPart(Situation1, Liam)" have been inferred in a similar manner. They have been caused by **Rule 6** and **Rule 5,** respectively. Rule 5 states that: *"In case a situation characterizes a memory and that memory is trying to be accessed by a person, than that person, as a contextual element, should be part of the same situation".* The properties "characterizes(Situation1, FfionsMemoryWithJonas)" and "wantsToAccess(Liam, FfionsMemoryWithJonas)" apply to the rule above. Also, **Rule 1** casts Liam as a

ContextualElement, as it states: *"In case a person would like to access a certain memory, then that person is a contextual element."*

Similarly, Rule 6 states that: *"In case a situation characterizes a memory and that memory is set in a particular location, then that location should be part of the situation (as a contextual element)".* The properties "characterizes(Situation1, FfionsMemoryWithJonas)" and "isSetIn(FfionsMemoryWithJonas, Location1)" apply to the rule above, causing the property "hasPart(Situation1, Location1)". As the previous example, Location1 is also classified as a contextual element, based on **Rule 2:** *"In case a memory is set in a certain location, then that location is a contextual element."*

Because of inversion, properties "partOf(Location1, Situation1)" and "partOf(Liam, Situation1)" have also been inferred.

The property "isNotOwnerOf(Liam, FfionsMemoryWithJonas)" has been inferred based on **Rule 4:** *"In case a person p1 would like to access a memory m1, a person p2 owns m1 and p1 and p2 are different people, then p1 is not the owner of m1 (since a memory can have only one owner)."* We have defined "wantsToAccess(Liam, FfionsMemoryWithJonas)" and "isOwnerOf(Ffion, FfionsMemoryWithJonas)". It was manually set in Protégé that Ffion and Liam are different individuals. Because of these three items, "isNotOwnerOf(Liam, FfionsMemoryWithJonas)" is inferred.

Lastly, we have "cannotAccess(Liam, FfionsMemoryWithJonas)" and "cannotBeAccessedBy(FfionsMemoryWithJonas, Liam)", which are inverse properties. For these inferences, a few rules were used. The main is **Rule 13**: *"There is a situation that currently characterizes the memory that is trying to be accessed by a person that is not the owner of the memory. If the situation is composed of a stressful emotional state, meaning, if the owner feels stressed about someone else trying to access their memory, then the person that wants access cannot access the memory".* The properties "characterizes(Situation1, FfionsMemoryWithJonas)" and "wantsToAccess(Liam, FfionsMemoryWithJonas)" were previously defined. Because of rule 4, the property "isNotOwnerOf(Liam, FfionsMemoryWithJonas)" was inferred. Also, there are the properties "feels(Ffion, Hesitant)" and "feels(Ffion, Nervous)", along with "experiences(Ffion, EmotionalState1)". Rule 8 inferred the property "hasPart(EmotionalState1, Hesitant)" and "hasPart(EmotionalState1, Nervous)". So, either **Rule 9**, *"In case an emotional state contains the psychological condition Nervous, then it is reclassified as a stressful state."* or **Rule 10,** *"In case an emotional state contains the psychological condition Hesitant, then it is reclassified as a stressful state."* classifies the EmotionalState1 as a StressfulState. Because of Rule 7, as previously explained, the property "hasPart(Situation1, EmotionalState1)" was inferred. So, Rule 13 is used: Situation1 characterizes FfionsMemoryWithJonas, which Liam, who is not the owner of the memory, is trying to access. Situation1 has EmotionalState1, which is also a StressfulState, meaning Ffion is stressed about Liam accessing her memory. Because of that, the property "cannotAccess(Liam, FfionsMemoryWithJonas)" is inferred. Inversely, "cannotBeAccessedBy(FfionsMemoryWithJonas, Liam)" is also inferred.

Also, a different classification of individuals has been inferred. Table 7 identifies what individuals have been classified as a new type and new types they represent.

Individual Name	New Type
Location1	ContextualElement
Liam	ContextualElement
IndividualMemory	Focus
EmotionalState1	ContextualElement
EmotionalState1	StressfulState

Table 7 - Individuals and their inferred types for the first scenario using MyMemory

Figure 5 illustrates Protégé's interface in case of inferences. What has been inferred is highlighted in yellow, to differentiate from manual definitions.

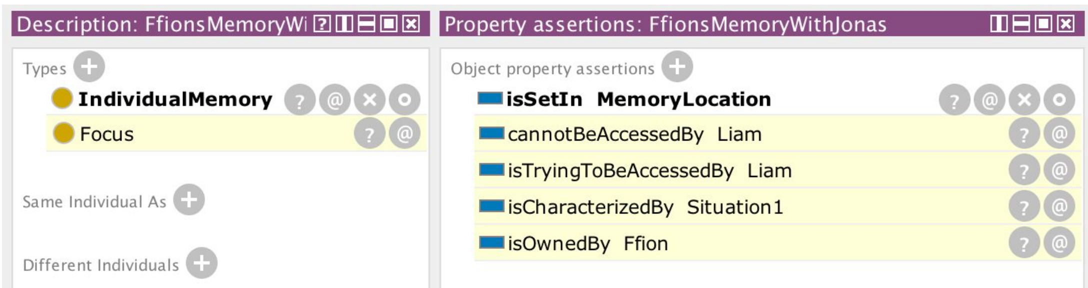

Figure 5 - Protégé screenshot showing the inferred new type of FfionsMemoryWithJonas and its inferred object property for the first scenario using MyMemory

4.1.4 Result analysis

Our rules defined that if someone wants to see someone else's memory, but the owner feels stressed, then the memory shouldn't be shared with the person that is requesting it. We believe that the owner's current emotional state, after a particular memory of him/her has been requested, is directly related to the person that is requesting and to the content of the memory. So, in case of a stressful emotional state, the memory should be kept private and not be sharable with whoever is requesting it.

So, in this scenario, Liam wants Ffion to show him her memory of her having sexual relations with Jonas during their marriage. However, because Ffion feels nervous and hesitant, and because Liam is not the owner of the memory, then Ffion won't be able to share the memory with him because the system will block such action in order to protect its user's privacy.

4.2 Scenario 2 - Public memory

This second example is a made-up situation.

4.2.1 Scenario description

In this scenario, Jonas, the man Ffion had an affair with, is with Ffion and asks to see their memory together - the same memory Liam wants to see in the previous example. The two of them are talking about their relationship and Ffions feels happy.

4.2.2 Ontology instantiation

To illustrate this scenario with our ontology, we have created the individuals as demonstrated in Table 8.

Individual Name	Type
Ffion	Person
Jonas	Person
FfionsMemoryWithJonas	IndividualMemory
EmotionalState2	EmotionalState
Happy	PsychologicalCondition
Location1	Location
Bristol	City
England	Country
Situation2	Situation

Table 8 - Individuals and their types for the second scenario using MyMemory

Figure 6 is a screenshot taken from Protégé regarding the individuals mentioned above.

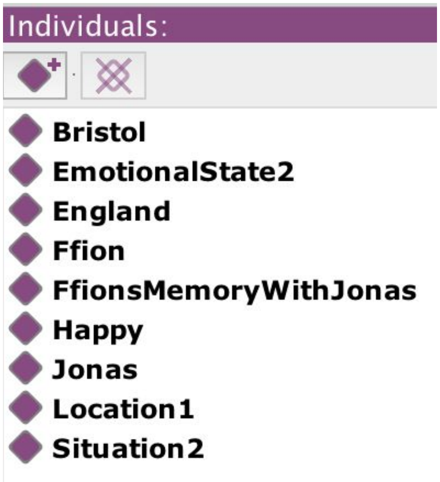

Figure 6 - Protégé screenshot of individuals for the second scenario using MyMemory

Next, Table 9 relates the individuals to the object properties we manually defined in Protégé.

Object property
feels(Ffion, Happy)
isOwnerOf(Ffion, FfionsMemoryWithJonas)
experiences(Ffion, EmotionalState2)
wantsToAccess(Jonas, FfionsMemoryWithJonas)
isSetIn(FfionsMemoryWithJonas, Location1)
hasPart(Location1, Bristol)
isLocatedIn(Bristol, England)
characterizes(Situation2, FfionsMemoryWithJonas)

Table 9 - Object properties for the second scenario using MyMemory

As it's possible to observe, Table 9 is very similar to Table 5, from the previous scenario. The main difference is Ffion's feelings - she no longer feels nervous or hesitant, she is now happy. Also, we changed the Individual from Liam to Jonas, but this alteration is merely semantic, it will not affect the outcome. Our rules don't consider an individual's name, for our ontology is meant to be used by anyone.
The individual FfionsMemoryWithJonas looks the same as Figure 4 from the first example, as it should - both examples have the same memory as its focus.

4.2.3 Ontology use

Once we synchronize the reasoner, we can see a difference in the inferences. Unlike the previous example, the memory now **can** be accessed by the person that is requesting it - in this case, Jonas. Because of the way Ffion is feeling, which is happy, or relaxed, the ontology detects that there is no problem in sharing a memory that is being talked about in the current moment. Because its owner, Ffion, feels relaxed, she is not worried about her privacy - she trusts the person that is requesting it.
Table 10 shows inferences that were made in object properties that differ from the first example. Some inferences stay the same, since only Ffion's emotional state changed.

Inferred Object Property
hasPart(EmotionalState2, Happy)
canBeAccesedBy(FfionsMemoryWithJonas, Jonas)
canAccess(Jonas, FfionsMemoryWithJonas)

Table 10 - Inferred object properties for the second scenario using MyMemory

The property "canAccess(Jonas, FfionsMemoryWithJonas)" was inferred similarly to the previous scenario. **Rule 12** states that: *"There is a situation that currently characterizes the memory that is trying to be accessed by a person. If the situation is composed of a relaxed emotional state, meaning, if the owner feels relaxed about someone trying to access their memory, then the person that wants access can access the memory."* It also used **Rule 11**: *"In case an emotional state contains the psychological condition Happy, then it is reclassified as a relaxed state."*

4.2.4 Result analysis

One thing that is important to note is that EmotionalState2 was reclassified as type **RelaxedState**. Ffion's feelings composed her emotional state in a way it is now classified as a relaxed state. Our rules state that if the person feels happy, when they feel relaxed. If they feel relaxed about someone else accessing a particular memory, then there is nothing to be hidden or marked as private. So, because Jonas is part of the reason why she is happy, he can have access to her memory.

Figure 7 shows Protégé's interface of FfionsMemoryWithJonas after the reasoner was synchronized, showing the inferences that happened based on our rules.

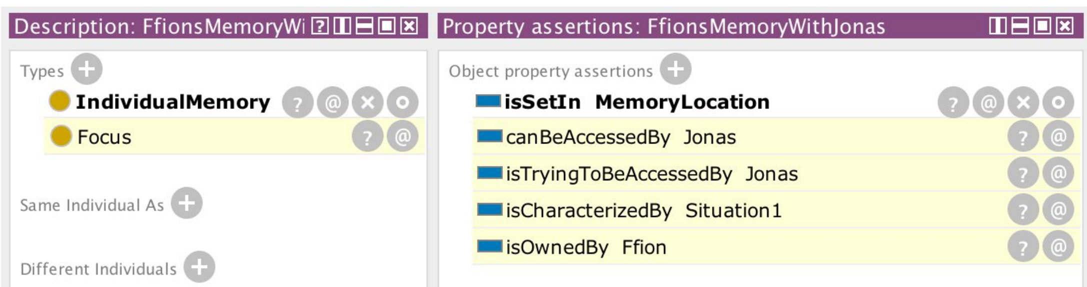

Figure 7 - Protégé screenshot showing the inferred new type of FfionsMemoryWithJonas and its inferred object property for the second scenario using MyMemory

4.3 Limitations

Each example fell into a different rule that determined whether the person (Liam or Jonas) could access Ffion's memory. The first example used the rule in which, if the owner feels "stressed" and if the person that was requesting access was not the owner him/herself, then they *should not* access the memory. The second example used the rule that, if the owner feels "relaxed" with the requester, then whoever is requesting *can* have access to the memory.

We have manually defined the set of rules, a set of EmotionalStates (StressfulState and RelaxedState) and a set of PsychologicalConditions (Nervous, Happy and Hesitant). Because of that, we based our examples on them. It's quite possible that a third example might not fall in any of those rules mentioned above. If this ontology was to be reused by a system without any modifications, it might work for some cases and not for others. There are infinite possibilities of how someone is feeling and infinite ways to classify someone's emotional state. We could not categorize and list each possibility, so we chose a few examples.

However, we are considering a system that is able to *learn* and *identify* new situations in which the user feels other psychological conditions, like anger or jealousy, so machine learning would be an important technology to be considered for implementation. Also, if the user denies access by screaming "NO" to whoever is requesting access, it should be enough for the system to detect a hostile situation. For that to happen, the system needs some sort of speech and also video recognition, which is not the focus of this work.

We have presented a way to incorporate privacy protection into the grain device seen on Black Mirror in [23] and more detailed here in this work, but it can be used by any system that manages memories. We believe our solution is a start to many other features regarding privacy, but still leaves room for other issues. Even though it is not the focus of this work, the machine learning capability we think the grain should have is not simple and requires a lot of work. It should be able to process speech and to interpret video in an extremely advanced way, so that it can think like its user. We visualize the grain, or whichever system, as being almost part of its user, who relies on its device for protecting their privacy.

4.4 Benefits

MyMemory is a semantically precise representation of the entities, properties and also situations involved in this domain of discourse. Other situations that may not fall in our rules can still have their entities/individuals matched to our classes. The semantic we have defined for each class is defined in Table 2, so it is a matter of applying the meaning to the system that would use our ontology.

Our ultimate goal is to protect people's privacy. We believe our work has some contributions to privacy protection when it comes to memories, for our definition of ContextualElements and Situation is what ultimately determines if a certain memory should be shared or not. Our rules which define ContextualElement can easily be implemented. For example, if a system that uses our ontology detects that somebody wants to access the user's memory, then that person becomes a ContextualElement, meaning it is an important variable to determine whether he/she should have access to the user's memory or not.

5 Conclusions

We have presented our extension of the External Digital Memory Ontology described in [19] with a focus in privacy protection.

Section 2 features an overview of topics we referred to a lot in this work. Section 2.1 presents an introduction on memory and its relationship with privacy. Section 2.2 focuses on Ontology, having three subsections: 2.2.1 states some characteristics of ontology languages; 2.2.2 discusses OWL and section 2.2.3 is about Protégé and the features we used for this work.

Section 3 is the formal definition of MyMemory. Section 3.1 analyses the original External Digital Memory Ontology, which is the base for our ontology. Section 3.2 presents our ontology and its characteristics, having four subsections: 3.2.1 describes the new classes created, 3.2.2 describes the new object properties, 3.2.3 is a small section about data properties and section 3.2.4 presents our rules, which ultimately define privacy.

Section 4 has examples of scenarios in which our ontology is used. Subsection 4.1 is an example of a situation in which a memory should be private, subsection 4.2 is an example of a memory being public, subsection 4.3 discusses some limitations of our ontology and subsection 4.4 regards its benefits.

We would like to stress that that this idea was based on "The entire history of you" and the grain, but could be reused and adapted by any information system that deals with memories or that could potentially have privacy invasion.

5.1 Limitations

Section 4.3 lists some technical limitations for MyMemory.
Also, our solution does not guarantee hacking attempts will not happen, which can also break users' privacy - even though "The entire history of you" does not touch on that subject. Furthermore, the grain's context interpretation is likely not going to be perfect. Its intelligence may also overcome the user's current state of mind. For example, if the user is intoxicated and wants to show private memories to other people in a context the grain may judge as not favorable, it could block said memories from being shown. This could cause trouble, but the grain would see it as protecting its user.
Another delicate example would be if the user was in trial for a certain crime. If the user was requested to give their memories as evidence, and if the user was guilty, the grain would sense the user is in trouble. However, we believe justice is a greater good, and the grain should not block the memories. This example illustrates how sensitive this topic is and how open it is for discussion, for we have not come to a full conclusion on what the best solution regarding privacy is. We have provided one in order to try to create space for other ideas that would enable humans to live a better future than the dystopian one presented in "The entire history of you".

5.2 Future work

MyMemory may have more possible contextual elements. We opted for three (the person that is requesting, how the owner is feeling and where the memory is located), but certainly there is room for more. Variables like "is the requester present in the memory?" or "action pattern in memory" do impact on whether the memory should be shared. In the example described in section 4.1, Liam wanted to see a memory of his wife having sexual relations with another man. The fact that Liam is *not* in the memory and that the memory contains *sexual relations* definitely affects Ffion's feelings, for it is such an intimate act with somebody else.

We expect a system that uses our ontology to have artificial intelligence capabilities. Speech and video recognition, for a system like the grain, could be extremely helpful to identify who is requesting the memory or how the user reacts to the request. Also, future work may regard how to store the information the system collects about the user's privacy wishes and feelings. Besides, it is also possible to explore the relationships among the different types of memories depicted in the External Digital Memory Ontology. For example, an important discussion would be about Collective Memory. Since one video contains an event with many participants, who is the owner (even if captured by only one individual)?

We could also extend the investigation to reflect on potential forms of abuse and discrimination that may stem from grain-like devices, e.g. the use of body-worn cameras by police. Different solutions are needed to maintain privacy in a world so public as the one portrayed in Black Mirror, in which people constantly share their memories and want to see other people's memories.

References

[1] Abburu, S. (2012). A Survey on Ontology Reasoners and Comparison. In *International Journal of Computer Applications*, pages 33-39.

[2] Brooker, C. (2011). Charlie Brooker: the dark side of our gadget addiction. Available at https://www.theguardian.com/technology/2011/dec/01/charlie-brooker-dark-side-gadget-addiction-black-mirror, last accessed 2017/11/15.

[3] Dentler, K., Cornet, R., Teije, A. and Keizer, N. (2011). Comparison of Reasoners for Large Ontologies in the OWL 2 EL Profile. In *Semantic Web Journal*, pages 71-87. IOS Press.

[4] Facebook Reports First Quarter 2017 Results, available at: https://investor.fb.com/investor-news/press-release-details/2017/Facebook-Reports-First-Quarter-2017-Results/default.aspx, last accessed 2017/11/15.

[5] Fensel, D., Harmelen, F. van, Horrocks, I., McGuinness, D.L. and Patel-Schneider, P.F. (2001) OIL: an ontology infrastructure for the Semantic Web, In *IEEE Intelligent Systems,* pages 38–45. IEEE.

[6] Gondar, J. (2008). Memória individual, memória coletiva, memória social. In *Morpheus - Revista Eletrônica em Ciências Humanas.* Available at: http://www4.unirio.br/morpheusonline/numero13-2008/jogandar.htm, last accessed 2017/11/15

[7] Gontier, E.M. (2015). Web Semantic and Ontology. In *Advances in Internet of Things (Volume 5, Number 2)*, pages 15-20. Scientific Research Publishing Inc.

[8] Gross, R. and Acquisiti, A. (2005) Information Revelation and Privacy in Online Social Networks(The Facebook case). In *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 71-80. ACM.

[9] Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. In *Knowledge Acquisition - Special issue: Current issues in knowledge modeling*, pages 199-220. Academic Press Ltd

[10] Guarino, N. (1998), Formal Ontology and Information Systems, IOS Press.

[11] Guarino, N., Oberle, D. and Staab, S. (2009). What Is an Ontology?. In: Staab S., Studer R. (eds) Handbook on Ontologies. International Handbooks on Information Systems, pages 1-17. Springer, Berlin, Heidelberg.

[12] Guizzardi, G., Baião, F., Lopes, M. and Falbo, R. (2010). The Role of Foundational Ontologies for Domain Ontology Engineering: An Industrial Case Study in the Domain of Oil and Gas Exploration and Production. In *International Journal of Information System Modeling and Design*, pages 1-22. IGI Publishing.

[13] Horrocks, I., Patel-Schneider, P. and Harmelen, F. (2003). In *Web Semantics: Science, Services and Agents on the World Wide Web*, pages 7-26. Elsevier.

[14] Le Goff, J. (1990), História e memória. Campinas: Editora da Unicamp.

[15] Mattos T.C., Santoro F.M., Revoredo K. and Nunes V.T. (2014). A formal representation for context-aware business processes. In *Computers in Industry (Volume 65, Issue 8)*, pages 1193-1214. Elsevier.

[16] Megill, A. (2004). History, Memory, Identity. In *Jeffrey K. Olick, Vered Vinitzky-Seroussi and Daniel Levy (eds.), The Collective Memory Reader*, pages 193-197. New York: Oxford University Press. (Excerpted from "History, Memory, Identity," History of the Human Sciences, August 1998, 11: 37-62).

[17] Protégé 5 Documentation - Class Expression Syntax. Available at: http://protegeproject.github.io/protege/class-expression-syntax/, last accessed 2017/11/15

[18] Pulido, J.R.G., Ruiz, M.A.G., Herrera, R., Cabello, E., Legrand, S. and Elliman, D. (2006). Ontology languages for the semantic web: A never completely updated review. In *Knowledge-Based Systems (Volume 19, Issue 7)*, pages 489-497. Elsevier.

[19] Rodrigues, J., Baião, F., Santoro, F. and Cappelli, C. (2017). Cyberspace: Towards an Ontology of the External Digital Contemporary Memory. In *Twenty-third Americas Conference on Information Systems*. AIS.

[20] Solove, Daniel J. (2008), Understanding Privacy, Harvard University Press, pages 1-4. Available at SSRN: https://ssrn.com/abstract=1127888, last accessed 2017/11/15

[21] Spiekermann, S. and Cranor, L.F. (2009). Engineering privacy. In *IEEE Transactions on Software Engineering (Volume 35, Issue 1)* pages 67-82 . IEEE.

[22] Studer, R., Benjamins, R. and Fensel, D. (1998). Knowledge engineering: Principles and methods. In *Data & Knowledge Engineering*, pages 161-198. Elsevier.

[23] Teixeira, B. R. ; Santoro, F.M. (2017). Memory and Privacy in The Entire History of You. In *Proceedings of the Re-coding Black Mirror 2017 Workshop co-located with 16th International Semantic Web Conference (ISWC 2017)*. Available online at: http://ceur-ws.org/Vol-1939/paper1.pdf, last accessed 2017/11/15

Appendix A - OWL code of MyMemory

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"

xml:base="http://www.semanticweb.org/bianca/ontologies/2017/9/MyMemor
y"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

ontologyIRI="http://www.semanticweb.org/bianca/ontologies/2017/9/MyMe
mory">
    <Prefix name=""
IRI="http://www.semanticweb.org/bianca/ontologies/2017/9/MyMemory"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf"
IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Declaration>
        <Class IRI="#StoredMemory"/>
    </Declaration>
    <Declaration>
        <Class IRI="#EmotionalState"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#relatedTo"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#canAccess"/>
    </Declaration>
    <Declaration>
```

```
    <ObjectProperty IRI="#isOwnedBy"/>
</Declaration>
<Declaration>
    <Class IRI="#StressfulState"/>
</Declaration>
<Declaration>
    <Class IRI="#Location"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#cannotAccess"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Hesitant"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#experiences"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Nervous"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#wantsToAccess"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Situation1"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#isNotOwnerOf"/>
</Declaration>
<Declaration>
    <Class IRI="#GeneratedMemory"/>
</Declaration>
<Declaration>
    <Class IRI="#Country"/>
</Declaration>
<Declaration>
    <Class IRI="#Happy"/>
</Declaration>
```

```
&lt;Declaration&gt;
    &lt;Class IRI="#Place"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;Class IRI="#SharedMemory"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;ObjectProperty IRI="#isLocatedIn"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;Class IRI="#Situation"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;Class IRI="#CulturalMemory"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;Class IRI="#ContextualElement"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;NamedIndividual IRI="#Liam"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;NamedIndividual IRI="#England"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;ObjectProperty IRI="#belongsTo"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;ObjectProperty IRI="#canBeAccessedBy"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;ObjectProperty IRI="#isSetIn"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;Class IRI="#ManipulatedMemory"/&gt;
&lt;/Declaration&gt;
&lt;Declaration&gt;
    &lt;ObjectProperty IRI="#isCharacterizedBy"/&gt;
```

```
    </Declaration>
    <Declaration>
        <DataProperty IRI="#meaning"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Nervous"/>
    </Declaration>
    <Declaration>
        <Class IRI="#MemoryData"/>
    </Declaration>
    <Declaration>
        <Class IRI="#CollectiveMemory"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#subsidizesChoice"/>
    </Declaration>
    <Declaration>
        <Class IRI="#SocialMemory"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Focus"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isFeltBy"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isTryingToBeAccessedBy"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#Bristol"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#feels"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ArtificialMemory"/>
    </Declaration>
    <Declaration>
```

```xml
    <NamedIndividual IRI="#Ffion"/>
</Declaration>
<Declaration>
    <Class IRI="#RelaxedState"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#isContentOf"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#characterizes"/>
</Declaration>
<Declaration>
    <Class IRI="#City"/>
</Declaration>
<Declaration>
    <Class IRI="#Person"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#partOf"/>
</Declaration>
<Declaration>
    <Class IRI="#RetrievedMemory"/>
</Declaration>
<Declaration>
    <Class IRI="#Hesitant"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#FfionsMemoryWithJonas"/>
</Declaration>
<Declaration>
    <Class IRI="#PsychologicalCondition"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#cannotBeAccessedBy"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#isOwnerOf"/>
</Declaration>
```

```xml
<Declaration>
    <NamedIndividual IRI="#EmotionalState1"/>
</Declaration>
<Declaration>
    <ObjectProperty IRI="#hasPart"/>
</Declaration>
<Declaration>
    <Class IRI="#BiologicalMemory"/>
</Declaration>
<Declaration>
    <Class IRI="#HistoricalMemory"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#Location1"/>
</Declaration>
<Declaration>
    <Class IRI="#IndividualMemory"/>
</Declaration>
<SubClassOf>
    <Class IRI="#ArtificialMemory"/>
    <Class IRI="#IndividualMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#BiologicalMemory"/>
    <Class IRI="#IndividualMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#City"/>
    <Class IRI="#Place"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#City"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#isLocatedIn"/>
        <Class IRI="#Country"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
```

```
    <Class IRI="#Country"/>
    <Class IRI="#Place"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#CulturalMemory"/>
    <Class IRI="#CollectiveMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#EmotionalState"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#hasPart"/>
        <Class IRI="#PsychologicalCondition"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#EmotionalState"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#relatedTo"/>
        <Class IRI="#IndividualMemory"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#EmotionalState"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#relatedTo"/>
        <Class IRI="#Person"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Focus"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#isCharacterizedBy"/>
        <Class IRI="#Situation"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Happy"/>
    <Class IRI="#PsychologicalCondition"/>
```

```xml
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Hesitant"/>
        <Class IRI="#PsychologicalCondition"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#HistoricalMemory"/>
        <Class IRI="#CollectiveMemory"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#IndividualMemory"/>
        <Class IRI="#Focus"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#IndividualMemory"/>
        <ObjectExactCardinality cardinality="1">
            <ObjectProperty IRI="#isOwnedBy"/>
            <Class IRI="#Person"/>
        </ObjectExactCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#IndividualMemory"/>
        <ObjectExactCardinality cardinality="1">
            <ObjectProperty IRI="#isSetIn"/>
            <Class IRI="#Location"/>
        </ObjectExactCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Location"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#hasPart"/>
            <Class IRI="#Place"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ManipulatedMemory"/>
        <Class IRI="#IndividualMemory"/>
    </SubClassOf>
```

```
<SubClassOf>
    <Class IRI="#Nervous"/>
    <Class IRI="#PsychologicalCondition"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#RelaxedState"/>
    <Class IRI="#EmotionalState"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#RetrievedMemory"/>
    <Class IRI="#IndividualMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#SharedMemory"/>
    <Class IRI="#IndividualMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Situation"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#hasPart"/>
        <Class IRI="#ContextualElement"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#SocialMemory"/>
    <Class IRI="#CollectiveMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#StoredMemory"/>
    <Class IRI="#IndividualMemory"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#StressfulState"/>
    <Class IRI="#EmotionalState"/>
</SubClassOf>
<ClassAssertion>
    <Class IRI="#City"/>
    <NamedIndividual IRI="#Bristol"/>
```

```
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Country"/>
        <NamedIndividual IRI="#England"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Person"/>
        <NamedIndividual IRI="#Ffion"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#EmotionalState"/>
        <NamedIndividual IRI="#EmotionalState1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#IndividualMemory"/>
        <NamedIndividual IRI="#FfionsMemoryWithJonas"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Hesitant"/>
        <NamedIndividual IRI="#Hesitant"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Person"/>
        <NamedIndividual IRI="#Liam"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Location"/>
        <NamedIndividual IRI="#Location1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Nervous"/>
        <NamedIndividual IRI="#Nervous"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Situation"/>
        <NamedIndividual IRI="#Situation1"/>
    </ClassAssertion>
    <DifferentIndividuals>
```

```
    <NamedIndividual IRI="#Ffion"/>
    <NamedIndividual IRI="#Liam"/>
</DifferentIndividuals>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#isLocatedIn"/>
    <NamedIndividual IRI="#Bristol"/>
    <NamedIndividual IRI="#England"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#experiences"/>
    <NamedIndividual IRI="#Ffion"/>
    <NamedIndividual IRI="#EmotionalState1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#feels"/>
    <NamedIndividual IRI="#Ffion"/>
    <NamedIndividual IRI="#Hesitant"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#feels"/>
    <NamedIndividual IRI="#Ffion"/>
    <NamedIndividual IRI="#Nervous"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#isOwnerOf"/>
    <NamedIndividual IRI="#Ffion"/>
    <NamedIndividual IRI="#FfionsMemoryWithJonas"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#isSetIn"/>
    <NamedIndividual IRI="#FfionsMemoryWithJonas"/>
    <NamedIndividual IRI="#Location1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#wantsToAccess"/>
    <NamedIndividual IRI="#Liam"/>
    <NamedIndividual IRI="#FfionsMemoryWithJonas"/>
</ObjectPropertyAssertion>
```

```
    <ObjectPropertyAssertion>
        <ObjectProperty IRI="#hasPart"/>
        <NamedIndividual IRI="#Location1"/>
        <NamedIndividual IRI="#Bristol"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty IRI="#characterizes"/>
        <NamedIndividual IRI="#Situation1"/>
        <NamedIndividual IRI="#FfionsMemoryWithJonas"/>
    </ObjectPropertyAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="#meaning"/>
        <NamedIndividual IRI="#Location1"/>
        <Literal
datatypeIRI="http://www.w3.org/2001/XMLSchema#string">couple&apos;s
house</Literal>
    </DataPropertyAssertion>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#cannotAccess"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isOwnedBy"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isTryingToBeAccessedBy"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
    <InverseObjectProperties>
        <ObjectProperty IRI="#belongsTo"/>
        <ObjectProperty IRI="#experiences"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#canAccess"/>
        <ObjectProperty IRI="#canBeAccessedBy"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
```

```
    <ObjectProperty IRI="#cannotAccess"/>
    <ObjectProperty IRI="#cannotBeAccessedBy"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#characterizes"/>
    <ObjectProperty IRI="#isCharacterizedBy"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#feels"/>
    <ObjectProperty IRI="#isFeltBy"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasPart"/>
    <ObjectProperty IRI="#partOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#isOwnedBy"/>
    <ObjectProperty IRI="#isOwnerOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#isTryingToBeAccessedBy"/>
    <ObjectProperty IRI="#wantsToAccess"/>
</InverseObjectProperties>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#belongsTo"/>
    <Class IRI="#EmotionalState"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#canAccess"/>
    <Class IRI="#Person"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#cannotAccess"/>
    <Class IRI="#Person"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#characterizes"/>
    <Class IRI="#Situation"/>
```

```
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#experiences"/>
        <Class IRI="#Person"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#feels"/>
        <Class IRI="#Person"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isContentOf"/>
        <Class IRI="#MemoryData"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isLocatedIn"/>
        <Class IRI="#City"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isNotOwnerOf"/>
        <Class IRI="#Person"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isOwnerOf"/>
        <Class IRI="#Person"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isSetIn"/>
        <Class IRI="#IndividualMemory"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#relatedTo"/>
        <Class IRI="#EmotionalState"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#subsidizesChoice"/>
        <Class IRI="#Focus"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
```

```
    <ObjectProperty IRI="#wantsToAccess"/>
    <Class IRI="#Person"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
    <ObjectProperty IRI="#belongsTo"/>
    <Class IRI="#Person"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#canAccess"/>
    <Class IRI="#IndividualMemory"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#cannotAccess"/>
    <Class IRI="#IndividualMemory"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#characterizes"/>
    <Class IRI="#IndividualMemory"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#experiences"/>
    <Class IRI="#EmotionalState"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#feels"/>
    <Class IRI="#PsychologicalCondition"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#isContentOf"/>
    <Class IRI="#IndividualMemory"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#isLocatedIn"/>
    <Class IRI="#Country"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#isNotOwnerOf"/>
    <Class IRI="#IndividualMemory"/>
```

```
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#isOwnerOf"/>
    <Class IRI="#IndividualMemory"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#isSetIn"/>
    <Class IRI="#Location"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#relatedTo"/>
    <ObjectUnionOf>
        <Class IRI="#IndividualMemory"/>
        <Class IRI="#Person"/>
    </ObjectUnionOf>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#subsidizesChoice"/>
    <Class IRI="#ContextualElement"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#wantsToAccess"/>
    <Class IRI="#IndividualMemory"/>
</ObjectPropertyRange>
<DisjointObjectProperties>
    <ObjectProperty IRI="#canAccess"/>
    <ObjectProperty IRI="#cannotAccess"/>
</DisjointObjectProperties>
<DisjointObjectProperties>
    <ObjectProperty IRI="#isNotOwnerOf"/>
    <ObjectProperty IRI="#isOwnerOf"/>
</DisjointObjectProperties>
<DataPropertyDomain>
    <DataProperty IRI="#meaning"/>
    <Class IRI="#Location"/>
</DataPropertyDomain>
<DataPropertyRange>
    <DataProperty IRI="#meaning"/>
```

```
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DLSafeRule>
    <Body>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#h"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Happy"/>
            <Variable IRI="urn:swrl#h"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#EmotionalState"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
    </Body>
    <Head>
        <ClassAtom>
            <Class IRI="#RelaxedState"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#h"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Hesitant"/>
            <Variable IRI="urn:swrl#h"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#EmotionalState"/>
```

```
                <Variable IRI="urn:swrl#eS"/>
            </ClassAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#StressfulState"/>
                <Variable IRI="urn:swrl#eS"/>
            </ClassAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Person"/>
                <Variable IRI="urn:swrl#p"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#belongsTo"/>
                <Variable IRI="urn:swrl#eS"/>
                <Variable IRI="urn:swrl#p"/>
            </ObjectPropertyAtom>
            <ClassAtom>
                <Class IRI="#Person"/>
                <Variable IRI="urn:swrl#p2"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#IndividualMemory"/>
                <Variable IRI="urn:swrl#iM"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#PsychologicalCondition"/>
                <Variable IRI="urn:swrl#pC"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#feels"/>
                <Variable IRI="urn:swrl#p"/>
                <Variable IRI="urn:swrl#pC"/>
            </ObjectPropertyAtom>
```

```
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#wantsToAccess"/>
            <Variable IRI="urn:swrl#p2"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#EmotionalState"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#isOwnerOf"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#relatedTo"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#p2"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#ContextualElement"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#relatedTo"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#pC"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
```

```
    <Body>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#characterizes"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Situation"/>
            <Variable IRI="urn:swrl#s"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#relatedTo"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#EmotionalState"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#eS"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#eS"/>
            <Variable IRI="urn:swrl#n"/>
```

```
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#EmotionalState"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#Nervous"/>
            <Variable IRI="urn:swrl#n"/>
        </ClassAtom>
    </Body>
    <Head>
        <ClassAtom>
            <Class IRI="#StressfulState"/>
            <Variable IRI="urn:swrl#eS"/>
        </ClassAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#characterizes"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Situation"/>
            <Variable IRI="urn:swrl#s"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#isSetIn"/>
            <Variable IRI="urn:swrl#iM"/>
            <Variable IRI="urn:swrl#l"/>
        </ObjectPropertyAtom>
        <ClassAtom>
```

```
            <Class IRI="#Location"/>
            <Variable IRI="urn:swrl#l"/>
        </ClassAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#l"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#isSetIn"/>
            <Variable IRI="urn:swrl#iM"/>
            <Variable IRI="urn:swrl#l"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Location"/>
            <Variable IRI="urn:swrl#l"/>
        </ClassAtom>
    </Body>
    <Head>
        <ClassAtom>
            <Class IRI="#ContextualElement"/>
            <Variable IRI="urn:swrl#l"/>
        </ClassAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ClassAtom>
```

```
            <Class IRI="#Person"/>
            <Variable IRI="urn:swrl#requester"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#isOwnerOf"/>
            <Variable IRI="urn:swrl#owner"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#wantsToAccess"/>
            <Variable IRI="urn:swrl#requester"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Person"/>
            <Variable IRI="urn:swrl#owner"/>
        </ClassAtom>
        <DifferentIndividualsAtom>
            <Variable IRI="urn:swrl#owner"/>
            <Variable IRI="urn:swrl#requester"/>
        </DifferentIndividualsAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#isNotOwnerOf"/>
            <Variable IRI="urn:swrl#requester"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="#Person"/>
```

```
            <Variable IRI="urn:swrl#p"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#characterizes"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#relatedTo"/>
            <Variable IRI="urn:swrl#rS"/>
            <Variable IRI="urn:swrl#p"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Situation"/>
            <Variable IRI="urn:swrl#s"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#rS"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#RelaxedState"/>
            <Variable IRI="urn:swrl#rS"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#wantsToAccess"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#canAccess"/>
```

```
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="#Person"/>
            <Variable IRI="urn:swrl#p"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#characterizes"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Situation"/>
            <Variable IRI="urn:swrl#s"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#StressfulState"/>
            <Variable IRI="urn:swrl#sS"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#isNotOwnerOf"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#sS"/>
        </ObjectPropertyAtom>
```

```
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#wantsToAccess"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#relatedTo"/>
            <Variable IRI="urn:swrl#sS"/>
            <Variable IRI="urn:swrl#p"/>
        </ObjectPropertyAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#cannotAccess"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="#Person"/>
            <Variable IRI="urn:swrl#p"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#characterizes"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
        <ClassAtom>
            <Class IRI="#Situation"/>
            <Variable IRI="urn:swrl#s"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
```

```
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#wantsToAccess"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Body>
    <Head>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#hasPart"/>
            <Variable IRI="urn:swrl#s"/>
            <Variable IRI="urn:swrl#p"/>
        </ObjectPropertyAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
    <Body>
        <ClassAtom>
            <Class IRI="#Person"/>
            <Variable IRI="urn:swrl#p"/>
        </ClassAtom>
        <ClassAtom>
            <Class IRI="#IndividualMemory"/>
            <Variable IRI="urn:swrl#iM"/>
        </ClassAtom>
        <ObjectPropertyAtom>
            <ObjectProperty IRI="#wantsToAccess"/>
            <Variable IRI="urn:swrl#p"/>
            <Variable IRI="urn:swrl#iM"/>
        </ObjectPropertyAtom>
    </Body>
    <Head>
        <ClassAtom>
            <Class IRI="#ContextualElement"/>
            <Variable IRI="urn:swrl#p"/>
        </ClassAtom>
    </Head>
</DLSafeRule>
<DLSafeRule>
```

```
        <Body>
            <ClassAtom>
                <Class IRI="#Person"/>
                <Variable IRI="urn:swrl#p"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#IndividualMemory"/>
                <Variable IRI="urn:swrl#m"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#wantsToAccess"/>
                <Variable IRI="urn:swrl#p"/>
                <Variable IRI="urn:swrl#m"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isOwnerOf"/>
                <Variable IRI="urn:swrl#p"/>
                <Variable IRI="urn:swrl#m"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#canAccess"/>
                <Variable IRI="urn:swrl#p"/>
                <Variable IRI="urn:swrl#m"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
</Ontology>



<!-- Generated by the OWL API (version 4.2.8.20170104-2310)
https://github.com/owlcs/owlapi -->
```