TECHNICAL REPORT IIIMTR-20240401
APRIL || 2024 || V1.0

SocialInsight: AN AGENT-BASED SIMULATION FRAMEWORK FOR POLICY CREATION & RESEARCH ON SUBSTANCE ABUSE PREVENTION IN YOUTH – A PROOF-OF-CONCEPT PROTOTYPE

ICELANDIC INSTITUTE FOR INTELLIGENT MACHINES

IN COLLABORATION WITH REYKJAVIK UNIVERSITY
&
ICELANDIC CENTER FOR SOCIAL RESEARCH & ANALYSIS

Kristinn R. Thórisson,[1,5] Sofia Basílio,[1] Gonçalo Carvalho,[1] Clark Upton Meyers[2]
Bridget Burger,[1,3] Sigurjón Guðjónsson,[3] Kári Hlynsson,[1] Inga Dóra Sigfúsdóttir,[3,4]
Caine Meyers,[3,4] Álfgeir Kristjánsson,[3] Stefán Ólafsson[5] & Marta Abad Torrent[1]

[1]Icelandic Institute for Intelligent Machines
[2]University of Iceland
[3]Reykjavik University, Department of Psychology
[4]Icelandic Center for Social Research & Analysis
[5]Reykjavik University, Department of Computer Science

Contents

1 Introduction **3**

2 LIFECOURSE **4**

3 Related Work **4**
3.1 Agent-Based Modeling & Simulation 4
3.2 Ontologies 6
3.3 Substance-Use Ontologies 7
3.4 Combining Ontologies & Agent-based Models 8

4 *SocialInsight:* Requirements **9**
4.1 Ontology 10
4.2 Agent-Based Framework 10
4.3 Scenarios 10
4.4 Ontology Interface 11
4.5 Interfacing Mechanism for Experiment Creation 12
4.6 Layers of Abstraction 12
4.7 Output Interface 13
4.8 Level of Analysis and Experimentation 14
4.9 Data Input 14
4.10 Risk-Protective Factors in the Framework 14
4.11 Scenario Templates 14
4.12 Validation and Scenario Preview 15
4.13 Detailed System Description for Scenario Design and Parameter Control . 15
4.14 Example 16
4.15 Model Verification and Validation 16
4.16 Technological Frameworks 18

5 *SocialInsight* Prototype Development **18**
5.1 Approach 18
5.2 Tools & Technologies 19
5.3 Upper Ontology Structure & Principles 21
5.4 Domain-specific Ontology Structure 25

6 Evaluation **32**
6.1 Adaptability 32
6.2 Extensibility 33
6.3 Accessibility 33
6.4 Scenario Implementation 33
6.5 Usability of the Present Prototype 34
6.6 Scalability 35

7 Conclusions & Future Work **36**

Executive Summary

Social phenomena represent a complex subject for scientific study due to the inherent complexities of human behavior, societal structures, and community dynamics. Such interplay is often multifaceted, exhibiting stochastic properties, and consequently making their modeling and replicability particularly challenging and resource-intensive, if not unfeasible.

The *SocialInsight* framework introduces a state-of-the-art approach to the design and evaluation of social intervention programs. The framework integrates advanced computation models and methodologies, resting on knowledge from across a range of disciplines within the social and life sciences, to supercharge how social programs and policies are created, studied and implemented, by governmental and academic institutions.

A key innovation of this framework is the transparency and management of extensible, scalable agent-based models that can dynamically adapt to new data and insights about the intricate, multifaceted real-world phenomena they represent. This adaptability is crucial in the realm of public policy and social programming, where understanding the nuanced interplay of sociological, psychological, and environmental factors is essential.

Underpinning SocialInsight is the combination of empirical research and advanced technical methodologies. The framework rests on the LIFECOURSE research results, a longitudinal study over 20 years of the effectiveness of youth substance abuse prevention programs. This foundation supports the scientific study of causal links influencing adolescent behaviour, as well as tailored approaches to intervening on such behavior at multiple levels, across a diverse range of contexts. An agent-based modeling framework, unified with an ontological content management, allows scientists and policy makers to keep track of much more complex systems than before. The framework is the key to more powerful multidisciplinary collaboration, fostering a broader and more unified understanding of human social behaviors, allowing policymakers and researchers to evaluate proposed social policies before their actual implementation.

The anticipated benefits of SocialInsight include bolstering the effectiveness of program and policy implementation, reducing associated costs and risks, and fast-tracking the pipeline between policy development and implementation. This report details the technological underpinnings of SocialInsight, as well as its ontological infrastructure, emphasizing the importance of a robust and flexible ontology system that can handle the complexity and variability inherent in modeling social systems.

The full scope of the project will involve improvements on all fronts of this proof-of-concept prototype, including scientific modeling, simulation comparison and management, collaboration features for distributed teams, and deeper analysis and modeling tools. In the closer future the project will focus on fine-tuning and increasing the scope of this proof-of-concept, expanding its capability to include a broader range of social phenomena, as well as continuing to bridge the gap between theoretical research and practical policy implementation. A graphical user interface for the numerous levels of use and system function must be developed.

This work was made possible in part by funding from the European Union, Reykjavik University and the Icelandic Institute for Intelligent Machines.

1 Introduction

This report describes the SocialInsight framework,[1] part of the SIMLife project at Reykjavik University and IIIM, that aims to bring to maturity a new computational framework and methodological approach for answering practical questions about the design and implementation of social intervention programs. The project represents a paradigm shift in the study and development of social programs and policies for both governmental and academic institutions.

The *SocialInsight* software[2] brings new tools for transdisciplinary collaboration across several fields of theoretical research, allowing experts in psychology, biology and sociology to work together to build deeper and broader models of human social behavior, and lays the groundwork for policymakers to be able to develop and evaluate policies before they are implemented, increasing speed and transparency of complex policy design while reducing associated risk and cost.

SocialInsight sits on two main pillars. Firstly, the LIFECOURSE research provides a theoretical foundation for the focus domain – substance abuse – resting on decades of data about the effectiveness of various intervention programs in a variety of social environments. By revealing the causal links between key factors that influence adolescent behavior with respect to substance consumption, a foundation is laid for an explicit computational approach that can be used to drive simulations of social mechanisms.

Secondly, *SocialInsight* adopts a modern approach to agent-based modeling and simulation (Abar et al., 2017; Bonabeau, 2002; da Silva, 2023; Thórisson et al., 2009; Thórisson et al., 2016), ontological knowledge networks (Livet et al., 2008; Phan et al., 2010) and causal-chain analysis (Baumann et al., 2020; Halpern & Pearl, 2013; Peters et al., 2020) to produce transparent, explainable models that expose theoretical mechanisms behind population dynamics and human behavior. The framework is organized around a comprehensive ontology management system (Thórisson et al., 2010) that captures both the topic domain and the computational constructs in a unified manner, enabling collaboration that spans several disciplines.

SocialInsight will allow improved development and testing of social policies and tactical initiatives prior to their launch, providing a unified scientific framework based on the best available data and expert knowledge, thus reducing their cost, uncertainty, and time-to-launch.

The SIMLife project will supercharge prior efforts and results in substance abuse prevention, modernizing existing development methods and knowledge and bring them into the 21st century. The work has potential to significantly impact three distinct domains: (a) Scientific research and social modeling, (b) public policy making and evaluation, and (c) social science education.

[1] This work is funded in part by the European Union ERC Proof-of-Concept grant LIFECOURSE-ABM (#101069400).

[2] The software is released under an open-source license (share-alike Creative Commons). https://github.com/IIIM-IS/SocialInsight-ABMS-SimulationOfSocialSystems – *accessed Oct. 23, 2023.*

2 LIFECOURSE

For decades scholars have been calling for a multilevel analysis of the bio-psycho-social nature of risk and protection of adolescent health and behavior (Kristjansson et al., 2010; Sigfusdottir et al., 2004). To do so, key information needed to be integrated so that uniform concepts could be established. Starting in 2015, the LIFECOURSE project[3] presented the world's first solution to this problem (Thorisdottir et al., 2020).[4] LIFECOURSE was the first study to include both refined measurements of environmental influences and sophisticated measurement of biological processes. Studies of early-life stress, in relation to the social environment and human biology, were conducted in disciplinary isolation. Before LIFECOURSE, much of the pre-existing knowledge had been segregated into separate disciplines, creating disciplinary silos. With LIFECOURSE this knowledge segregation was removed. LIFECOURSE presented a new comprehensive approach to address the influence of stress on adolescent behaviors, including substance use, suicidal behaviour, self-harm, and delinquency.

Since its launch, LIFECOURSE has resulted in multiple cross-disciplinary collaborations, and numerous questions have already been answered about the stress-behavior pathways and the role of the mediators and moderators in these relationships. The work models the nature of the bio-social link between stress, emotions, and behavior; the mediating and moderating effects of multiple environmental factors during specific developmental periods, whether these effects are cumulative across periods, and finally, whether these effects could be protected against or even reversed. The implications of LIFECOURSE are unprecedented, generating completely new knowledge about the bio-social pathways to adolescent behaviors across multiple disciplines, merging them into one.

3 Related Work

We consider related work in two main topics: Agent-based models and simulations, on the one hand, and ontological research on the other. The last section reviews work on their combination.

3.1 Agent-Based Modeling & Simulation

ABM applications have been used in various fields, including biology, psychology, sociology, and cross-disciplinary studies. Beheshti and Sukthankar (2014) developed an agent-based model to examine the influence of social norms and various interventions on smoking cessation trends. The authors structure their agents around three distinct classes of factors influencing smoking behavior: personal-values, social-networks, and

[3]Funded in part by a €2 million European Research Council Consolidator grant.

[4]The LIFECOURSE team is lead by Dr. Sigfúsdóttir and includes researchers at Reykjavik University's (RU) Dep. of Psych. and Dep. of Comp. Sci., the Icelandic Centre for Social Research and Analysis (ICSRA), and several overseas collaborators at King's College (UK), Columbia University (US), and elsewhere (Kristjansson et al., 2021; Kristjansson et al., 2010; Kristjansson et al., 2013; Sigfusdottir et al., 2004; Sigfúsdóttir et al., 2009).

environmental-influences. Personal-values are specific to individuals and based on personality traits. Two of these values are derived from Schwartz's sociological theory of cultural value orientation (Schwartz, 2006), assessing an agent's susceptibility to feedback from others (embeddedness vs. autonomy) and their level of ambition, daring, and assertiveness (mastery vs. harmony). Other values considered at the personal level included regret in the context of smoking and addiction, concern for their health, and their propensity to seek pleasure. Social-networks follow the power law degree distribution wherein a small number of nodes, referred to as hubs, possess a disproportionately large number of connections, while the majority of nodes maintain relatively few connections. They also exhibit homophily characteristics, a sociological concept suggesting that nodes sharing similarities tend to gravitate towards one another. Environmental-factors influencing social networks can be classified into four categories: (1) exposure to the behaviors of others, (2) presence of signs or posters, (3) advertising, and (4) miscellaneous factors encompassing digital, educational, and promotional activities.

A more recent simulation model for New York City proposed to explore the relationship between alcohol outlet density and alcohol-related violence (Castillo-Carniglia et al., 2019). The model was designed to incorporate demographic characteristics, such as age, sex, and race, as well as behavioral factors, such as drinking status and geographic proximity to outlets. The drinking status of the simulated population was determined based on a combination of sociodemographic and neighborhood-level factors. The primary aim of the model was to evaluate the impact of targeted intervention strategies, aimed at reducing alcohol outlet density, on rates of alcohol-related violence. Specifically, the model explored scenarios in which overall alcohol density was capped or specific outlets associated with high levels of violence were targeted for control measures. The authors use ABMs to simulate various scenarios and interventions, analyzing their effects on smoking cessation trends over time, conducting a sensitivity analysis to explore the impact of different parametrization on the outcome. The model provides valuable insights into the interplay of various factors influencing smoking behavior, it shares a limitation with Scwartz' (2006) alcohol outlet density model: Both models fail to account for a number of factors relevant in the study of substance use intervention. Modifying the existing model to include additional factors would likely require resource-intensive re-structuring. Although the work provides many valuable insights, the authors acknowledged that the model did not fully account for the complex interplay of psychological, biological, and social factors that are known to play a part in alcohol consumption behaviors. While a limitation of the particular implementation and not the methodology per se – in fact, quite the contrary – it is common to see limitations along this dimension in many agent-based modeling implementations targeting cross-disciplinary phenomena.

A number of other models produced recently are subject to this limitation (Bobashev et al., 2018; Dray et al., 2008; Stankov et al., 2019) and while ABMs offer a promising approach for investigating complex topics, incorporating a comprehensive range of variables and interactions between agents remains a challenging endeavor. A multidisciplinary approach drawing on expertise from diverse fields is needed to advance the modeling process and enhance its accuracy and applicability. New methodologies are needed to address the numerous challenges such work faces.

Various toolkits have emerged in recent years that aim to simplify the implementation of agent-based models by enabling programmers to avoid developing mechanisms from scratch. These frameworks include popular toolkits such as Netlogo (Tisue & Wilensky, 2004), Swarm (Minar et al., 1996), Mason (Luke et al., 2004), and others. These toolkits were developed to cater to the diverse range of applications of agent-based models, with some targeting non-human structures such as bacterial populations, infrastructures, and information systems, while others focus on social and human phenomena such as the financial market, healthcare systems, and epidemics.

The Netlogo toolkit has been employed for developing two distinct models that focus on substance use, namely SimArc (Lamy et al., 2011) and SimUse (Lamy et al., 2015). Both models were developed by the same authors and consider the topics of alcohol abuse, and the trajectories of recreational poly-drug users, respectively. Notably, these models share a commonality not only in their focus on substance use but also in their consideration of the interplay between neuroscience and substance use. Although SimArc and SimUse are distinct models, they exhibit some degree of conceptual overlap in the constructs they aim to define. It is worthy to note that these models remain entirely disconnected from each other. One of the current challenges in implementing agent-based models using toolkits is the disconnection of models that exhibit overlapping constructs. In the context of social science models, the interplay of social interactions is a shared mechanism that is relatively consistent across various substances. Therefore, there is a need to explore avenues for integrating these shared mechanisms in order to develop more comprehensive and robust models.

The design methodology of ABMs may vary in the process; however, a critical aspect for achieving a satisfactory outcome is the collaboration with domain experts in the field of study being simulated. Generally, these experts are responsible for decomposing the underlying rules and principles of the system, thereby facilitating their translation into a computer-based model. This collaborative process often concludes once the model has been sufficiently refined, verified, and deemed to represent the phenomenon adequately. However, as previously noted, a primary limitation of this approach is the potential longevity deficit for the modeling solution, particularly in light of new information and insights in the models replicated. This constraint highlights the importance of incorporating model adaptability and extensibility mechanisms to ensure the computational representation's ongoing relevance and applicability. We propose separating phenomenon representation and model implementation to achieve adaptability and extensibility by using ontologies as input into the ABM framework.

3.2 Ontologies

Ontologies pertain to the formalization of a particular domain of interest, serving as a means to represent entities, properties, and interrelations. Ontology components can be categorized into classes, individuals, attributes, and relations. For example, *Fluffy* (Individual) is a type of *Dog* (Class) with brown fur and blue eyes (attributes) and is owned by (Relation) *Jim* (Individual). Domain representation through ontologies, using an appropriate syntax for structuring the information, yields a multitude of benefits, including (1) providing a shared foundation for discourse among individuals with diverse backgrounds and perspectives; (2) offering extensive standalone functionality while enabling compatibility with various systems; (3) facilitating the identification of required components for the development of new systems; and (4) ensuring that as the ontology expands, its reliability is bolstered, granted that newly incorporated elements do not compromise the existing structural integrity (Uschold & Gruninger, 1996).

Ontologies are typically separated into two parts: The *upper ontology* and the *domain-specific* part. The first refers to a model of high-level, general-level descriptive terms, the terms required to represent one or many domain concepts. The latter is a model of a specific domain of the world, utilizing the components mentioned above and others to describe knowledge. In some instances, particular domains may require the design of an upper ontology before being properly able to implement the domain-specific one (Chandrasekaran et al., 1999).The limitations of the ABMs discussed previously concern the development of a modular and adaptive tool where new knowledge is easily incorporated into the ABM. A potential pathway to achieve this is through ontologies, incorporating separately domain-specific knowledge, and using the ontology as input into an ABM framework.

The Resource Description Framework (RDF) is a widely-used standard for representing and exchanging ontology information on the Web in a serialized format. The RDF is based on a model of the world in which everything is represented as a “resource”, and relationships between resources are described using a set of triples, which consist of a subject, predicate, and object. In the RDF, resources are identified by Uniform Resource Identifiers (URIs), which provide a globally unique identifier for each resource. Relationships between resources are represented using predicates, which are also identified using URIs, and whose function is to describe the nature of the relationship between the subject and object of a triple. The RDF enables the encoding and reuse of structured metadata, providing a means for publishing both human-readable and machine-processable vocabularies designed to encourage the reuse and extension of metadata semantics among disparate information communities (Miller, 1998).

3.3 Substance-Use Ontologies

Ontologies exist on the domain of substances and substance use, and taking on different representational approaches, the main themes being the hierarchical classification of drugs, their ingredients, as well as their chemical, therapeutical, and addictive properties (Brown et al., 2004; Chen et al., 2012; Hanna et al., 2013; Lipscomb, 2000). As the primary objective for developing an ontology is to promote reusability, these ontologies were reviewed with the objective of researching their applicability in the context of ABM. As a result, the applicability of these ontologies for our specific purpose is considered limited, given that substantial restructuring would be necessary to employ them in our context. Nevertheless, they may prove to be valuable references for future endeavors aimed at enhancing the domain-specific ontology proposed in this thesis.

An example of a valuable reference for the upper ontology includes the Descriptive

Ontology for Linguistic and Cognitive Engineering (DOLCE).[5] Developed by the Laboratory for Applied Ontology (LOA) at the Institute for Cognitive Sciences and Technologies of the Italian National Research Council (CNR), DOLCE aims to facilitate the integration, interoperability, and reusability of domain-specific ontologies by offering a foundational vocabulary and structure (Borgo et al., 2022). DOLCE captures fundamental components to human cognition and language, while remaining as neutral as possible, by deriving most of its structure and components from philosophy and linguistics, with respect to specific domain theories. Our approach and DOLCE share similarities in the upper ontologies they propose, that is, in the abstraction from domain-specific theories. While DOLCE attempts to remain exempt from any human theory, our framework's goal is to remain as neutral from any human social phenomenon to be represented as possible. This is achieved by describing general terms that can represent a large number of social theories in the agent-based model space.

3.4 Combining Ontologies & Agent-based Models

A review of the main techniques, methodologies, and applications of agent-based systems in the context of ontology alignment and semantic interoperability is given by Davidovsky et al. (2012). The authors provide a good overview of the key advantages, such as the potential for increased adaptability, scalability, and robustness.

Socio-ecological systems (SeS) share similarities with the domain of susbtance use, as both encompass multiple disciplines and are well suited for the application of ontologies in structural modeling and in conjunction with ABMs, as suggested in a recent paper by Gotts et al. (2019). The authors propose the use of four distinct ontologies: *System*, *model*, *project*, and *representation*. These ontologies are designed to bridge the concepts of ABMs and the conceptual representation of a domain of discourse, which is similar to our approach. Additionally, we incorporate ideas from Thórisson et al. (2016) , who proposed the separation of expert-level ontologies from user-level ontologies; a combination of version-control and meta-ontologies to keep track of the evolution and drift of sub-ontologies.

Our *system* ontology is designed to capture entities, relationships, and processes present in a specific part of the world, drawing upon defined terms from the project ontology. The *model* ontology is intended to capture the different models considered for the same system at different levels of detail, or distinct perspectives.

The primary distinction between the approach proposed by e.g. Gotts et al. (2019) and our own lies in the number and combination of ontologies employed. In our approach, a separate ontology is used for the domain of discourse and its models, and another for the representation of ABM-related concepts.[6]

While there have been other papers on the topic of ontology and ABM integration, very few implementations of proposed approaches exist. In one of these implementations,

[5] See http://www.loa.istc.cnr.it/dolce/overview.html – *accessed Oct. 25th, 2023.*

[6] Future versions may represent mechanisms separately from these, which would lead to the emergence of additional ontologies. However, our current ontology approach adopts a depth-first strategy, which prioritizes the development of comprehensive encompassing ontologies.

the authors put forth an ontology-based approach to tackle the challenges associated with reusability, interoperability, and expressiveness in the domain of ABMs (Christley et al., 2004). The proposed ontology encompasses fundamental components such as agents, environments, processes, and parameters, aiming to offer a standardized and structured representation of the concepts, relationships, and processes inherent in ABMs. While the authors underscore the potential of ontologies in facilitating the development of models and simulations, a significant distinction between their approach and ours lies in our objective to facilitate collaboration with domain experts. Consequently, the ontology's structure must strike a balance between simplicity and comprehensibility while effectively representing the necessary concepts in an ABM.

4 *SocialInsight:* Requirements

Based on the review of existing techniques and taking into account the problems to be addressed, the following list of requirements was devised for the long-term vision of the proposed approach (in order of importance):

1. Empower domain experts researching complex (social) systems to focus on subsets of their research without losing sight of or abandoning its relationship with its superset, including the larger context and variables. This includes:
   (a) Promoting the reuse of mechanisms intrinsic to the study of various dependent variables, independent of the phenomenon under consideration.
   (b) Facilitating modifications to the factors considered in the simulation and their representation within the agent-based model (ABM), ensuring adaptability to evolving domain knowledge.
2. Enable system improvements (e.g. scalability, decentralization, communications) without interfering with domain-specific knowledge and representation.
3. Ensure a seamless and transparent translation of domain concepts to their corresponding representations in the ABM.
4. Give domain experts a new framework and tool for working on complex models of social phenomena, including
   (a) those with minimal knowledge of ABMs,
   (b) allow anyone, from novice to expert, to easily incorporate both hypothetical and empirical information pertaining to the target phenomenon,
   (c) provide a platform with the potential to produce a comprehensible visualization of domain information, and
   (d) minimize the learning curve for domain-experts to understand how their models of the world are employed within the tool.

To fulfill these requirements, modularity, adaptability, and expandability must be considered at various levels of detail. It is imperative that not only does the ABM itself exhibit these characteristics; the overall tool must embody these principles.

4.1 Ontology

As previously mentioned, ontologies can serve as valuable instruments in formalizing a specific domain of interest, providing a means to represent entities, properties, and interrelations, encouraging reuse and extension of knowledge represented among disparate information communities. They are well suited for the module responsible for describing a phenomenon; however, current ontological frameworks offer only a general nomenclature for representing a domain, such as classes, instances, properties, and interrelations. To depict a domain in the context of ABM, a structure tailored to accommodate ABM components is required; this would involve an upper ontology encompassing the definition of agents, environments, parameters, perceptions, decisions, actions, as well as their interplay.

With the addition of the upper ontology, the final designated modules in the approach are: (1) An upper ontology describing general terms encapsulating the components of an ABM; (2) a domain-specific ontology describing the phenomenon to simulate using the general terms from the upper-ontology; (3) an ABM framework that utilises the domain information and executes a simulation given pre-defined settings.

4.2 Agent-Based Framework

For the ABM framework to be capable of simulating the information in the domain-specific ontology, it needs (a) a component that parses information from the ontology, (b) a component that initializes agents required to run the simulation given that information and initialisation settings such as population demographics and other statistics, (c) a component that controls and executes the simulation.

A significant drawback of current ABM implementations is the manner in which models are adjusted in light of new information on the phenomenon of study. Rigid representations make such changes often too expensive or unfeasible, rendering the original model entirely obsolete. To improve the longevity of models and ease collaboration with experts, we integrate ontologies to keep track of all concepts, including modeling agents (simulation modules), factors, and interactions. Our approach to ontology construction and use is meant to ensure that the target phenomenon representation is separate from the model implemented, significantly increasing the longevity and flexibility of construction.

4.3 Scenarios

Through an analysis of the LIFECOURSE research project (see Section 2 above; Kristjansson et al., 2010; Sigfusdottir et al., 2004; Sigfúsdóttir et al., 2009; Thorisdottir et al., 2023) and consultation with domain experts, we identified a variety of sufficiently general target scenarios to simulate. While the data encompasses a wide array of dependent and independent variables, and their respective interactions, the central dependent variable we focus on in *SocialImpact* is of course *substance use*.

The identified scenarios include:

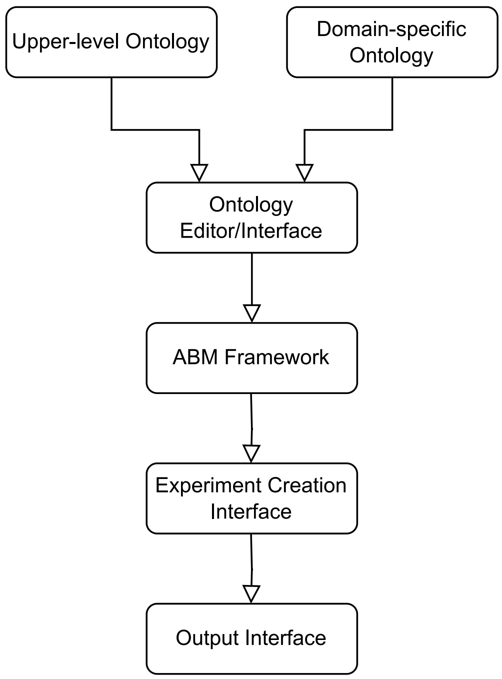

Figure 1: Ontologies, categorized into **upper-level ontology** and **lower domain-specific ontology**, serve as structured data repositories. The **Ontology Editor/Interface** acts as a mediator allowing users to tweak and communicate with these ontologies. The **ABM Framework** (Agent-Based Modeling framework), is the core simulation executor. Both the **Experiment Creation Interface** and the **Output Interface** are intrinsic parts of this framework, guiding users from experiment setup to result visualization.

1. Substance consumption, as the primary focus of the subject matter is substance use, the modeling of this core activity was a requirement.
2. Routines and commitments, chosen due to their association with delinquent behavior and the increased propensity for drug use.
3. Protective and risk factors, of varying importance, depending on the factors included, e.g. in the inclusion of both parental monitoring and support (the first has a stronger impact as a protective factor).
4. The effects of peer activities on individual behavior, this scenarios should target cases in which the perception of certain events or messages can directly affect an individual's propensity (eagerness or hesitancy) to perform some actions (e.g. yielding to peer pressure, resisting bullying, etc.).
5. Human plan-making. Individuals must be able to decide to make an action available when an opportunity arises, e.g. deciding to smoke when a cigarette becomes available. The scenarios must be sufficiently represented in several ways in the domain-specific ontology, using concepts from the upper ontology.

4.4 Ontology Interface

A graphical user interface (GUI) displays ontologies as node graphs, each node representing a unique entity, property, or relationship within the ontology. Users can click on a node to reveal an editing pane where they can modify attributes or add annotations. A Natural Language Processing (NLP) engine (or LLM) can be embedded into the GUI to interpret and convert textual inputs into changes in the ontology. For instance, if a scientist types "Add a relationship between X and Y" the NLP engine translates this into a new edge between nodes X and Y in the graph.

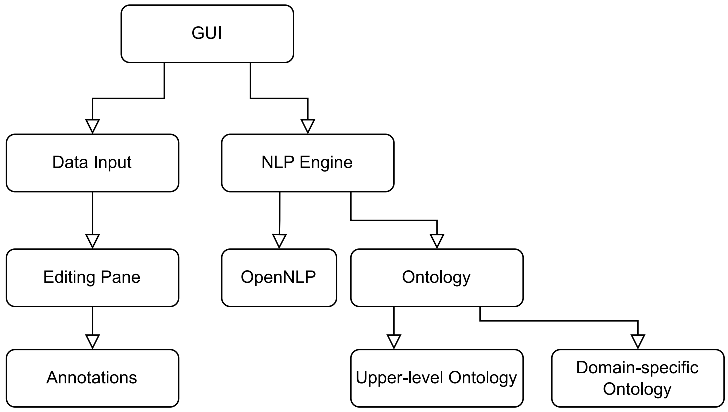

Figure 2: The graphical user interface (GUI) serves as a visual platform to display ontologies in the form of node graphs. Each **Node Graph** illustrates the ontologies and their relationships. Users can utilize the **NLP Engine** to transform textual descriptions into ontology modifications. To modify attributes or provide additional context, the **Editing Pane** and **Annotations** come into play. The **Ontology** holds structured information which can be broadly categorized into **Upper-level Ontology** and **Domain-specific Ontology** based on their specificity.

4.5 Interfacing Mechanism for Experiment Creation

An extension module within the GUI allows transitions from ontology editing to experiment creation. This module features an interface populated with policy templates that represent agent types, environment parameters, and interaction rules. These blocks are standardized but can be customized through editable fields. The interface allows users to create a simulation scenario by adding, removing and changing parameter values, essentially creating a storyboard for the simulation.

4.6 Layers of Abstraction

Users can switch between different layers of abstraction in both the ontology editor and the experiment creation interface. A toggle that lets users switch from a detailed view to an abstract view is available, where in the abstract view a complex set of interactions is represented as a single, editable “black box” component. As researchers gather more data or insights, they can revert to the detailed view to edit the internals of the “black box” component, but its specificity is not a requirement for simulating an environment.

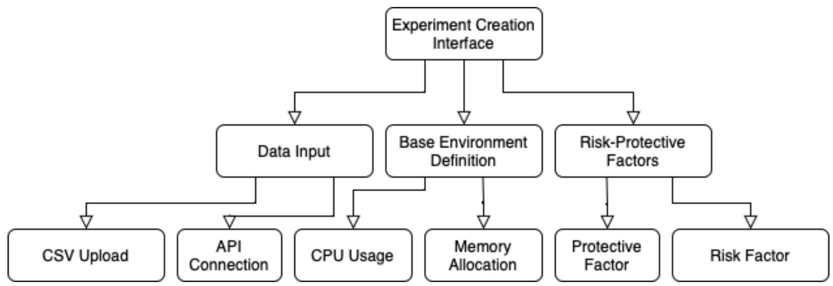

Figure 3: For initiating simulations, the **Data Input** section allows users to define starting conditions. The **Base Environment Definition** outlines the foundational environment attributes of the experiment. The **Risk-Protective Factors** help in identifying variables either as protective measures or potential risks. Users can introduce data through **CSV Upload** or real-time **API Connection**. The system's performance is monitored and adjusted through metrics like relaying **compute resources**. Within the risk domain, variables can be distinctly marked as **Protective Factor** or **Risk Factor**.

4.7 Output Interface

The output consists of a dashboard that displays real-time outputs of the simulation. Data visualization techniques are used to create dynamic graphs and heat maps, as well as statistical models that essentially report on the state of the simulated environment. The dashboard is interactive, enabling users to pause the simulation, modify parameters directly within the dashboard, and then continue the simulation. This allows for iterative modeling without requiring the user to be familiar with programming.

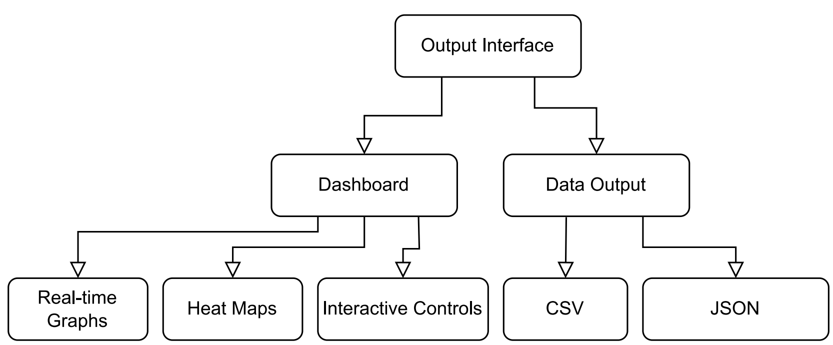

Figure 4: The **Dashboard** offers real-time outputs of the ongoing simulations. Under the **Data Output** umbrella, users can anticipate results in varied formats, one of which includes **Real-time Graphs** and **Heat Maps**. Furthermore, the system offers **Interactive Controls** for on-the-fly simulation modifications. Data can be represented in formats such as **CSV** and **JSON**.

4.8 Level of Analysis and Experimentation

Users are allowed to specify the granularity of data capture and analysis. This could range from high-level outputs like population trends down to micro-level data on individual agent actions. This feature is accessible both during the experiment setup phase and in the output interface.

4.9 Data Input

The data input module supports multiple formats. Users are able to input initial data conditions through a form-based interface, upload CSV files, or connect to an API for real-time data retrieval. A parser is available to convert this data into a format that can populate the initial conditions of the simulation.

4.10 Risk-Protective Factors in the Framework

Handling Risk-Protective Factors within the experiment creation interface enables users to specify variables that are recognized as protective or risk factors in the context of the simulation. For example, in a public health simulation, "vaccination rate" can be identified as a protective factor, while "population density" can be a risk factor for disease spread.

The module presents users with a table-like interface where each row represents a different variable in the simulation. Next to each variable, a checkbox provides options to mark it as either a *"Protective Factor "* or a *"Risk Factor"*. The table has additional columns where users can input numerical values or ranges that define the threshold levels for each factor.

Risk-Protective Factors Module

Once these factors are identified and their thresholds set, the framework automatically tags them within the simulation logic. During the simulation run, if a protective or risk factor crosses its defined threshold, an event trigger may be activated. This event trigger can generate alerts or modify other variables in the simulation.

Exportable Data

The data pertaining to these risk-protected factors is made easily exportable for further analysis. It is possible to download this data in various formats like CSV or JSON.

4.11 Scenario Templates

Scenario Templates provide structured starting points for creating experiments; they are designed by domain experts to encompass key variables and mechanisms often considered in specific types of social science or policy research.

Parameter Sliders

Upon selecting a template, the Parameter Sliders section appears with sliders and input fields for adjusting key variables in the template, offering a tactile and visually intuitive way to manipulate numerical parameters.

Property Editor

The Property Editor allows users to fine-tune parameters, providing contextual information and advanced settings.

4.12 Validation and Scenario Preview

Before script generation, the system conducts a validation check to ensure that all mandatory fields are populated and that templates have consistent parameters with real values.

Saving and Loading Scenarios

Users can save their workspace configurations for future use. The configurations are saved in an XML format, and a "Load Configuration" option allows users to upload a previously saved workspace.

4.13 Detailed System Description for Scenario Design and Parameter Control

Objectives

- Enable complex experiments in social sciences and policy analysis.
- Support variables with multi-level dependencies.

Functional Components

1. Scenario Templates: Predefined domain-specific templates.
2. Parameter Sliders: Manipulation controls for variables.
3. Variable Editor: Interface for adding and defining variables.
4. Validation & Dependencies Check: Ensures logical consistency.
5. Preview Scenario: Pre-execution review of the setup.
6. Scenario Outcomes: Real-time visualization and metrics.

Templates contain:

- Obligatory Independent Variables
- Base Environment Property
- Hypothesized Dependent Variables
- Initial Relationships

Parameter Sliders

Each obligatory independent variable has an associated slider. Features include manual setting, automatic setting based on historical data, and constraints.

Variable Editor

Capabilities include adding variables, defining relationships, and specifying whether variables are independent or dependent.

4.14 Example

Considering a policy experiment about 'Educational Outcomes' set in a 'School' environment, with base environment and variables defined alongside specified correlations, the system can generate scenarios that model these correlations and validate these hypotheses through automated metrics and validation techniques.

- Base Environment: School
- Variables: 'Teacher Quality,' 'Student Engagement,' 'Rate of Learning'
- Correlation: 'Teacher Quality' positively correlates with 'Student Engagement' (Coefficient = 0.7, Obligatory)
- 'Student Engagement' positively correlates with 'Rate of Learning' (Coefficient = 0.8, Hypothesized)

By specifying these relationships, the system can generate scenarios that model these correlations and can validate these hypotheses through automated metrics and validation techniques.

4.15 Model Verification and Validation

Our approach involves a process for developing ABMs targeting social phenomena at several levels of detail. The verification, validation, and testing (VV&T) processes of resulting AMBs must encompass them all, especially with a view on guaranteeing the longevity and adaptability of any generated simulations.

While ABM VV&T shares certain similarities with ontology VV&T, particularly in terms of ensuring that the outcome adequately addresses the questions proposed during the pre-design stage, for ontologies the validation and verification processes are distinct (Gómez-Pérez et al., 2006).

A VV&T process is required for three distinct parts: (1) upper ontology, (2) domain-specific ontology, and (3) ABM Framework. The initial iteration of any ontology-driven design approach concludes with the ABMS framework generating simulation data. Although the subsequent steps involve comparing the simulation data with available empirical data and obtaining expert verification of the simulation outcomes, it is expected that this often results in unsatisfactory outcomes. In these cases, a social simulation designer may call for a comprehensive review of the approach and background assumptions, addressing the following aspects:

1. **Upper ontology**: Does the upper ontology encompass the essential concepts to adequately represent the phenomenon under study? A thorough examination of the ODD protocol (**grimmeEtAl2020**) or other model description methods used is necessary to evaluate the generalization capabilities of the upper ontology concerning the subject matter. It is important to note that should the ontology lack specific terms to represent a particular effect, the initial assessment should focus on the model description, as some described mechanisms may be too specific to the phenomenon and could be alternatively represented through existing mechanisms represented in the current upper ontology version.
   VV&T Method: Taxonomy evaluation through the check of inconsistencies, incompleteness, and redundancies of concepts included in the ontology (Lovrencic & Cubrilo, 2008), and in comparison with respect to existing ABM components required to model the phenomenon through the ODD protocol or any other model descriptions utilized.

2. **Domain-specific ontology**: Are the theoretical models within the target phenomenon accurately represented using the most appropriate terms or mechanisms? The representation of certain effects using the upper ontology terms warrants a review to determine if it is the most suitable way to depict the effect or the emergence of an effect.
   VV&T Method: In consultation with domain experts, checking the ontology against real world representation and language. Consistency checks of similar interactions and their representations. Sensitivity checks to small changes in definition, ensuring the rest of the representations do not collapse (Lovrencic & Cubrilo, 2008).

3. **ABM Framework**: Similar to contemporary ABM implementations, the ABM Framework must be assessed for correctness on multiple levels for its underlying mechanisms. Should the framework behave as intended, model tuning may be required to fit the phenomenon and behaviors expected. That is, for the action value calculated in the decision-making process, the fine-tuning and balancing of action component importance and values should be addressed.
   VV&T Method: Formulation of test cases to evaluate the model based on the sequence in which processes are executed. Verification of emergent behavior through the analysis of observed macro-level outcomes and ensuring these are not preprogrammed or explicitly defined. Ensuring agent heterogeneity reflected in population demographics and input through the simulation initialization parameters. Corroborating information flows in a decentralized manner, such that agents and processes do not rely on a central source for information. Model is robust under various conditions such as changes in parameters, initial conditions, or agent behaviors to assess if the outcome remains consistent with the expect behavior (Balci, 1998).

The proposed VV&T steps (1) and (3) should be conducted whenever new concepts are introduced to the upper ontology. That is, following the addition of new concepts to the ontology, data retrieval and manipulation methods must be added to the ABM before the new concepts are usable. Step (2) should be carried out in an iterative manner when representing a social phenomenon, ensuring that the concepts are accurately represented using the upper ontology concepts. Step (3) should also be conducted when the ABM framework is being used without any changes to the rest of the components.

4.16 Technological Frameworks

To address the requirements above, technologies are needed for:

- Encoding ontological structures, e.g. Notation3 (N-Triples), Terse RDF Triple Language (Turtle), JavaScript Object Notation for Linked Data (JSON-LD), and Resource Description Framework/XML (RDF/XML).
- Faciliating the extraction of pertinent information from the ontologies, through query languages like RDF Query Language (RQL), SeRQL, Triple, and SPARQL.
- Storing and visualize the ontologies for example through graph database systems like GraphDB, Neo4J, and Titan-Cassandra.
- Programming the ABM framework, predicated on the established efficacy needed for this domain. Most commonly utilized languages include Python, C++, and Java.
- Facilitating the parsing and effective use of the ontologies developed and compatible with the programming language chosen, such as RDFLib for Python, Raptor RDF Syntax Library for C++, and Apache Jena for Java.

5 *SocialInsight* Prototype Development

The *SocialInsight* system was created with modularity, adaptability, and expandability in mind. It consists of three key components: An **ontology** (composed of an *upper* and *lower domain-specific* one), an **ABMS framework**, and a **scenario and runtime management system**.[7] Our approach for using these in a longitudinal simulation-building effort, for scientific and/or practical purposes, can be considered a fourth component, although a methodological one.

The following sections encompass the current state of the approach implementation, including (1) the upper ontology and all of the structures devised to represent an ABM, (2) the utility of the domain-specific ontology in representing scenarios in social phenomena, using examples from the dataset aforementioned on substance use among youth, (3) a complete documentation on the current ABM Framework and its composition.

5.1 Approach

A systematic process was employed in the development of *SocialInsight*, beginning with: (1) identifying the problem, challenges or limitations the approach aims to address,

[7] The scenario and runtime management system is currently under development (July 26, 2023) while initial versions of the other two components have already been produced.

(2) reviewing existing methodologies and key implementations within the domain, (3) defining a list of requirements and objectives, including encompassing outcomes, scope, and target audience.

Our approach addresses two key requirements:

I. To facilitate the continued collaboration among a diverse range of experts from various disciplines, each contributing unique insights to the target social phenomenon under investigation.
II. To address the limitations of current ABM implementations, as previously noted in Section 3, particularly with respect to the integration of new information and the potential resource-intensive nature of adjustments, which may not be feasible, given the complexity of the overall systems.

The approach taken to develop the prototype is illustrated in Figure 5, commences with the establishment of general descriptive terms that encapsulate the fundamental components of an agent-based model (ABM), encompassing the delineation of agents, environment, perceptions, decisions, and actions. It is imperative to note that the prevalent functions of an ABM are preserved within the codebase; however, the agent, perception, action, and other constituent elements subject to simulation are derived from the domain-specific ontology. This ontology, in turn, employs the general terms originating from the upper ontology as a bridge to facilitate seamless integration with the ABM framework. The ABM framework consists of three primary components, each of which plays a critical role in the simulation process. The first step involves parsing the types of agents, their attributes, behaviors, and relationships from the domain-specific ontology. Once this data has been parsed and all of the agent types and elements have been created, they are passed to the Sim-Initializer component, which instances the agents and the environment. Lastly, the Sim-Controller component takes in the instances and runs a simulation for a pre-defined number of iterations, producing data on the status of the population at the end of the simulation. This data is evaluated in the verification process to assess the accuracy and results through comparison with empirical data, and assessed by domain experts.

5.2 Tools & Technologies

The implementation of the approach relied on several key tools and technologies, given the integral involvement of ontology development and an ABM framework in the process.

We use RDF and Turtle syntax for encoding ontologies, GraphDB for managing and querying RDF data, Java for programming the ABM framework, and Apache Jena to load data in the framework from the ontologies. We describe the assumptions for these choices below. Steps to verify, validate, and test the approach are presented at the end.

The selection of the requisite tools and technologies was grounded in a set of carefully considered criteria (see Section 4). Primarily, their conventional usage and historical precedence in the context of the approach was a determinant factor. This criterion was predicated on the premise that the long-established tools have already proven their efficacy and robustness in similar applications, ensuring reliability and accuracy in the

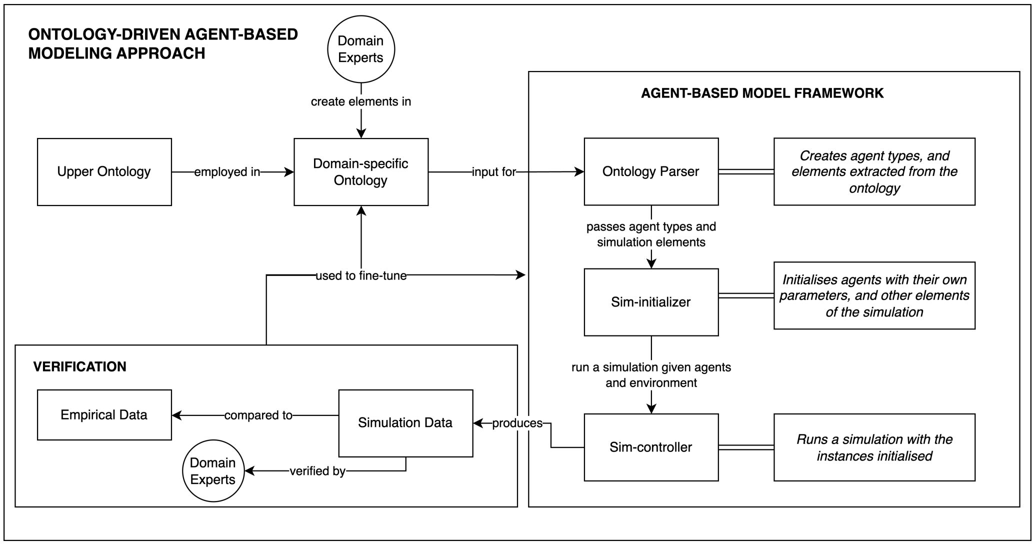

Figure 5: llustration of the iterative process involved in the ontology-driven agent-based modeling approach. The upper ontology is first designed to generate general descriptive terms, which are then employed in the creation of a domain-specific ontology that describes the phenomenon of interest. The Ontology-Parser component in the ABM Framework parses the information contained in the domain-specific ontology to identify element types that are to be simulated. The Sim-Initializer component instantiates instances of the elements previously created and passes them to the Sim-Controller for simulation. Following the completion of the simulation, the resulting data is outputted and compared against empirical data and by domain experts. Feedback from the verification process is used to fine-tune the domain-specific ontology and ABM framework, with the aim of improving the accuracy and effectiveness of the simulation.

present context. Furthermore, these tools and technologies were chosen due to their well-documented nature and the relative ease of application in the current thematic context, ensuring a smooth operational process, minimizing potential bottlenecks associated with complexities in tool deployment or incompatibilities with the existing system. Finally, a significant consideration in the selection process was the potential for seamless integration of the chosen tools and technologies with the existing modules, as well as their adaptability to accommodate future modules. This foresight was fundamental to future-proofing the system, ensuring its sustainability, and maintaining flexibility in accommodating technological evolution or enhancements to the approach.

The Resource Description Framework (RDF) was selected as the fundamental framework for conceptual representation in the development of the ontologies. This choice stems from its extensive and enduring utilization, as well as its provision of numerous essential properties and relationships. The Turtle syntax was chosen for encoding the relationships between concepts in the form of triples for numerous reasons including human-readability, compatibility with RDF, expressive power, and its tool support.

GraphDB was chosen to store, manage and query the RDF data as it is readily accessible and provides a comprehensive overview and detailed assessment of the ontologies, allowing for the communication of progress on the ontologies through visual and interactive graphs, and enabling figures to be effortlessly produced for this thesis.

Java was selected as the programming language for the ABM framework due to its object-oriented nature, which inherently aligns with the structure of ABMs (Railsback & Grimm, 2019), providing a robust and flexible podium for simulation implementation, as well as fostering the development of a modular and scalable framework. Additionally, Java features Apache Jena (McBride, 2001), a well-documented and consistently updated library for ontology management. Apache Jena facilated the creation of an RDF model within the ABM framework, enabling the loading of information from the ontologies and the execution of queries for relevant ABM components using the SPARQL query language, which is also employed in GraphDB. Facilitating the development, verification and testing of query outcomes by first previewing the query results in GraphDB, and subsequently transferring these queries for use by Apache Jena.

In conclusion, the selection of tools and technologies was based on a careful consideration of historical precedence, well-documented applications, ease of use, and adaptability, thereby ensuring a robust, reliable, and flexible implementation of the approach.

5.3 Upper Ontology Structure & Principles

The development of the upper-ontology followed an iterative and systematic approach. The initial step involved the formulation of a highly general term serving as the superclass for all other classes. Subsequently, the focus shifted towards determining the representation of agents and identifying the essential properties required for the initial iteration of our approach. Furthermore, careful consideration was given to discerning the types of agents that were fundamental and applicable across diverse simulation themes grounded in social phenomena.

The first term in the upper ontology is *Thing*, which serves as the most general and all-encompassing term from which other classes may inherit. Consequently, it functions as the superclass for all classes within the ontology. As of the current version of the upper ontology, there are no other terms between *Thing* and *Agent*, however, in the future more terms can and may be added. *Agent* is a subclass of *Thing* possessing properties such as *Parameter*, *Perception*, and *Action*. Parameters embody the variables intrinsic to an agent, including characteristics like age, height, and any other properties that potentially impact the agent's behavior. *Parameter* is also formulated as a class, encompassing minimum and maximum value, and name as its constituent properties. Parameters are instrumental in the agent-based modelling of a phenomenon, as they not only affect the agent's behavior during runtime but also contribute to the delineation of population composition. This distinction of populations is, in part, derived from the unique characteristics of the constituent individuals.

In our upper ontology, agents are currently further divided into *Autonomous* and *Non-Autonomous*. The reason for this distinction lies in the types of actions these agents may implement and the fact that in many ABMs, infrastructures and objects may be considered to be agents. Autonomous agents may change their location, acquire and consume resources, and in future iterations implement other actions that humans or animals may implement. Figure 6 illustrates the structure of components mentioned to this point, as well as an example of how *Human*, *Object*, and *Place* classes may be added through the domain-specific ontology.

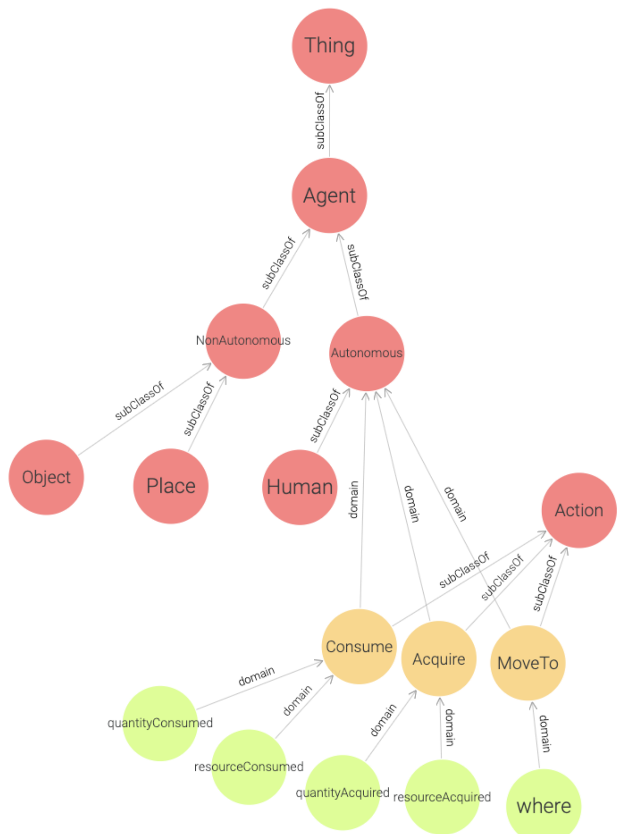

Figure 6: High-level structural representation of agents in the upper ontology, accompanied by example usage of the representation of agent types *Human*, *Object*, and *Place* within the domain-specific ontology. Arrows denote the existence of a relationship from a subject (originating) to an object (terminating), with the relationship specified along the arrow. Red signifies classes, yellow are subclasses of the class Action, and green indicates properties.

The next key term to describe are the available actions to the agent. Actions refer to the behaviors or activities agents may perform within the simulation environment. The selection of an action (decision-making) is guided by the internal state of an agent, previously perceived information, and interactions with other agents. In light of the substantial interdependence between actions and decision-making processes, decisions are not delineated as a distinct term. Rather, decision-making is encapsulated by the components considered in the computation of the final action value. The action value, in turn, serves as the determining factor in selecting the appropriate action for execution at a given time, and is a function of the following components:

- Satisfied pre-conditions: The availability of certain actions requires the fulfillment of specific pre-conditions, although the action may not be available at that time, due to its high action value, the agent may choose to satisfy the pre-conditions (or wait for them to be satisfiable, in the case of time) , with the plan of making the action available for execution. Pre-conditions can be of type location, time, and resource (e.g., execute the action of *BUY_DRINK* the agent must be located within a store, and to move to the store, the store must be open).
- Parameter factors and coefficients: Factors that are parameters of the agent which influence the tendency (positively) or hesitancy (negatively) to implement an action (e.g., the act of greeting another agent may be contingent upon the current mental state, wherein anxiety may exert a negative influence on predisposition), these are paired with assigned coefficients in the domain-specific ontology.
- State factors and coefficients: Antecedent events, or actions that have produced a state may influence the propensity to execute an action; similar to factors, these are paired with coefficients in the domain-specific ontology.
- Commitment: Commitments are obligations or promises to oneself or other agents. In the proposed framework, these commitments are temporally bounded, the execution of the associated action must be accomplished within a specified time frame while restraining the implementation of any potentially conflicting actions until the designated period has elapsed (e.g., the action of moving to school is tied to the commitment of attending school, adherence to the commitment entails refraining from movement to locations outside the school premises during the class period). Commitments may be pre-defined or initialised at run-time.

It is noteworthy that, while these components contribute to the final action value, not all actions require pre-conditions, factors, states, or commitments in their composition. However, to enhance the likelihood of its selection from the available set of actions, it is advisable for the action to be influenced by at least one of these components, as it would in a real-world scenario. This consideration ensures a more comprehensive and dynamic representation of agent behaviors within the simulation environment, accounting for the diverse factors that may impact decision-making processes. A complete graph of the components pertaining to the class Action can be found in Figure 7.

Presently, the upper ontology delineates three distinct categories of actions: (1) Movement, (2) Resource Management, and (3) General, these categories differ in the type of state produced and state manipulation as follows:

1. **Movement**: an agent changes their position within the simulation space, resulting in a new location state.
2. **Resource Management**: the acquisition and consumption of resources previously

acquired; these items are stored in the resource state. For an item to be consumed, it must reside in the resource state (pre-condition); upon executing the consumption action, the item's quantity is reduced by some value indicated in the domain-specific ontology.

3. **General**: any action which is not of type (1) or (2). These actions may be communication with other agents or any others considered in the phenomenon of study, the states produced by these actions are inserted into a history stack of states.

In addition to being influenced by various elements, actions may also yield effects upon execution. As previously noted, these effects may encompass the generation of new states or modification of pre-existing ones. Furthermore, actions may produce new commitments and emit messages (perceptions) to proximate agents or designated cohorts within the simulation environment.

As previously described, parameters serve as crucial components in ABMs, significantly influencing the agents' decision-making process and subsequent behavior. Within our framework, each agent's parameters are assigned an initial value prior to the start of the simulation, and these values may undergo alterations during run-time. Such modifications transpire due to known interrelations between parameters and other parameters, the perception of messages, and the execution of actions. Consequently, the upper ontology conceptualizes the following relationship pairs: parameter-parameter, perception-parameter, and action-parameter.

Parameter-parameter relationships entail that a variation in one parameter (the subject) influences the value of another parameter (the object) via a specific function. Within the domain-specific ontology, any given parameter may possess a relationship with numerous other parameters governed by distinct functions. Parameters can thus maintain relationships characterized by an ordered sequence of objects and functions. In these functions, keywords such as SCURRENT, SPREVIOUS, and OCURRENT may be employed, representing the subject's (the parameter that has the relationship to others) current value, the subject's previous value, and the object's (parameters in the object sequence) current value, respectively. Furthermore, mathematical operators are permissible within these functions, which are expressed using post-fix notation to maintain a rigorous representation.

Perception-parameter relationships entail that when an agent receives a certain message described in the domain-specific ontology, the parameters associated in the relationship will change by some function, much like in the parameter-parameter relationship, the functions are written in post-fix notation, but the only keyword that can currently be employed is OCURRENT, this is because perceptions do not have a value attached to them as of the current implementation.

Action-parameter relationships are triggered when an action is executed, where the objects and functions are described in the same way as perception-parameter relationships. The following Figure 8 was taken using GraphDB upon loading the upper ontology, where objects are a sequence (rdf:Seq) of Parameter, and functions are a sequence of String.

Following the annotation of the aforementioned general terms, the subsequent phase

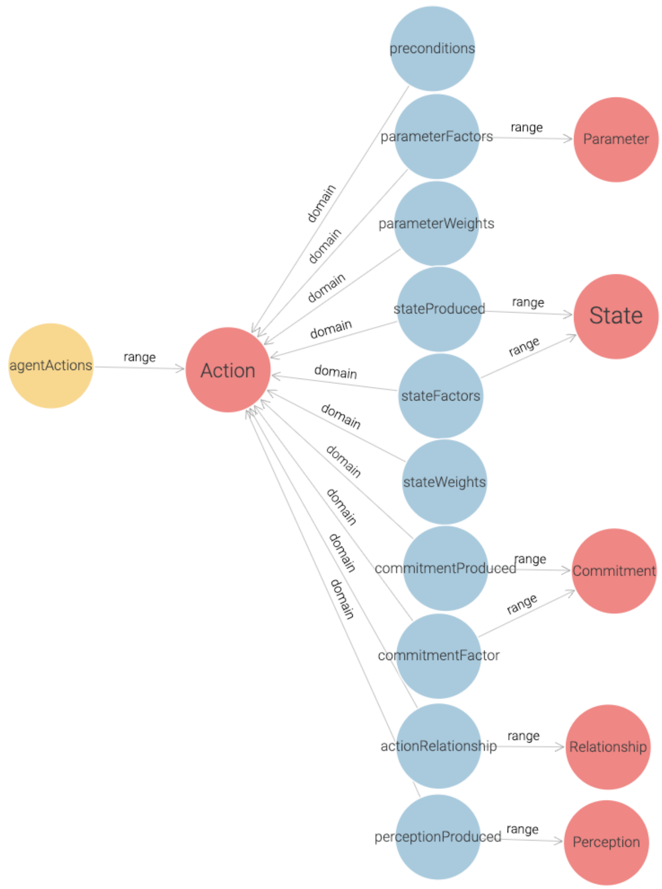

Figure 7: Complete list of properties of the class *Action* in the upper ontology. Blue circles are the properties of the class *Action*, red circles signify the class type the property belongs to, and yellow is the property containing a sequence of actions, namely agent actions is a list of *Action*.

involved was constructing a domain-specific ontology that employed these terms to represent a particular domain.

5.4 Domain-specific Ontology Structure

Prior to employing the domain-specific ontology, it is crucial to comprehensively examine the target phenomenon to determine its representation within an ABM. This can be accomplished by utilizing the ODD protocol, which provides a structured approach for assessing and describing the essential components of the model. Once the structure of the ABM has been decided, the first step is determining whether the agents specified belong

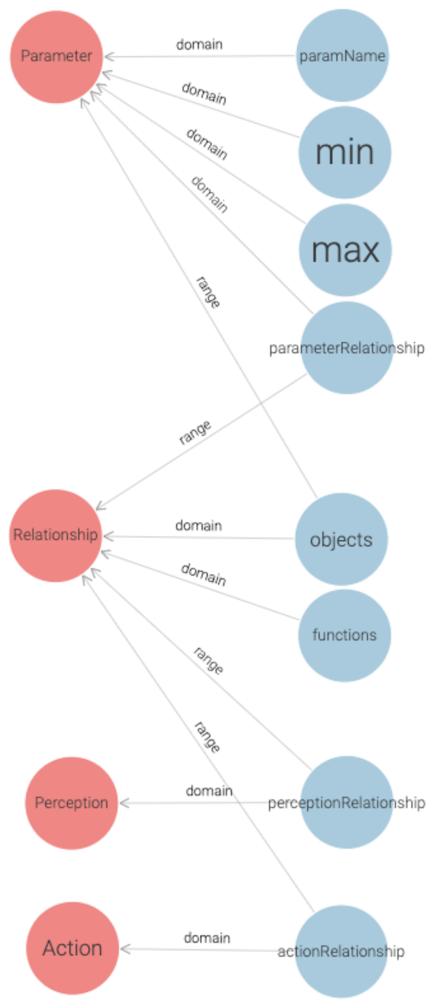

Figure 8: Illustration of the connections between parameter and parameter-parameter (paramRelationship), perception-parameter (perceptionRelationship), and action-parameter (actionRelationship) relationships. Red circles signify classes. The label range on the arrow signifies that the subject (originating) is of the variable type of the object (terminating), and domain indicates that the subject is an attribute/property of the object.

to the Autonomous or Non-autonomous classes represented in the upper ontology. For example, one may opt to represent locations as agents within the simulation. In this case, *Place* would serve as a class, inheriting from the *Non-autonomous* class since locations cannot alter their position or acquire or consume resources. Within the context of substance use among youth, an example of such a place is a school. Schools would inherit from the *Place* class and possess actions, such as RING_BELL, which are triggered upon perceiving specific times of the day (e.g., '8 a.m.' for when class starts or noon for lunch breaks), this example is illustrated in Figure 9.

Substance use among youth represents a sufficiently specific domain, highlighting the significance of representing youth for example through the *Child* agent type. Although this agent type could be a direct subclass of *Autonomous*, it may be more appropriate to designate *Child* as a subclass of *Human*, which, in turn, is a subclass of *Autonomous*. This distinction might be made to facilitate the creation of other agent types that share some of the attributes (Parameters, Perceptions, Actions) present in *Child*, but are not of the type *Child* or inherit from it. For that reason, the domain experts may opt to create the *Human* class, assigning the attributes specific to *Human* and then subsequently to *Child*. An instance of this distinction, illustrated in Figure 10, and in the context of substance use are the Parameter, Perception, and Action pertaining to smoking, such as propensity_to_smoke, smoking, and CONSUME_CIGARETTE respectively. A child may inherit those attributes from *Human*, and may have other specific to *Child*, such as the parameter grades, and the action move_to_school with a commitment tied to attending classes.

In the context of the substance use example, we might want to represent the relationship between a child's frequency of attending sports activities and their propensity to smoke (Kristjansson et al., 2021). Using the general terms from the upper ontology, this relationship can be represented in two distinct ways within the domain-specific ontology:

1. If the designer chooses to include propensity to smoke as an attribute, this relationship can be represented as a parameter-parameter relationship, as illustrated in Figure 11. In this case, we would create two parameters in the domain-specific ontology: sports frequency and propensity to smoke. The parameter relationship would be assigned to sports frequency, with the object list containing propensity to smoke, and the functions list incorporating a function representing the effects of sports frequency on propensity to smoke. Subsequently, propensity to smoke would act as a parameter factor in the action CONSUME_CIGARETTE.
2. Alternatively, if the designer opts not to include propensity to smoke as an attribute, sports frequency would directly serve as a parameter factor with a corresponding coefficient in the action CONSUME_CIGARETTE allowing in turn for other parameter factors to be included in the calculation of the action value for CONSUME_CIGARETTE, as seen in Figure 12.

Option #1 may prove beneficial for representing bivariate models, while option #2 is better suited for multivariate models. It is essential to note that directly inputting values from empirical data into these models may result in biased outcomes. The primary objective of employing these different relationship representations is to replicate similar effects observed in empirical data, such as those found in bivariate and multivariate models.

Perception relationships can be employed to represent the effects that arise when an agent perceives a particular event or behavior. In the context of substance use, one frequently examined variable is the number of peers who engage in substance use around the individual under investigation (Kristjansson et al., 2013). This interaction can be con-

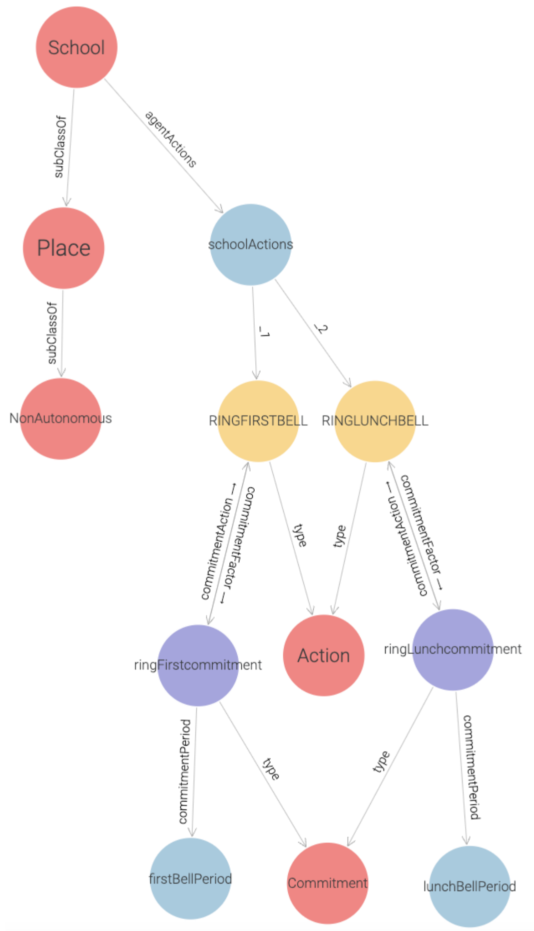

Figure 9: How to represent School in the domain-specific ontology as an agent type, inheriting specifically from *Place* which in turn inherits from *NonAutonomous*, where school contains actions RINGFIRSTBELL (ringing the bell at 8 a.m.) and RINGLUNCHBELL (ringing the bell at noon for lunch). To enforce that this action is chosen, a commitment is added for the periods at which these actions should be executed.

ceptualized as the effect of perceiving someone the agent is acquainted with, partaking in activities such as smoking cigarettes, consuming alcohol, or using other substances.

In the ontology, we can represent the perception *smoking* as a type of perception that is generated when an agent carries out the action CONSUME_CIGARETTE and is subsequently perceived by neighboring agents. We would then create *smoking* as an instance of perception and associate it with the perception produced by the action CONSUME_CIGARETTE. This perception would possess a perception relationship where the objects list contains the parameter it influences, such as the parameter *propensity_to_smoke*. Consequently, an agent whose friends frequently smoke in their

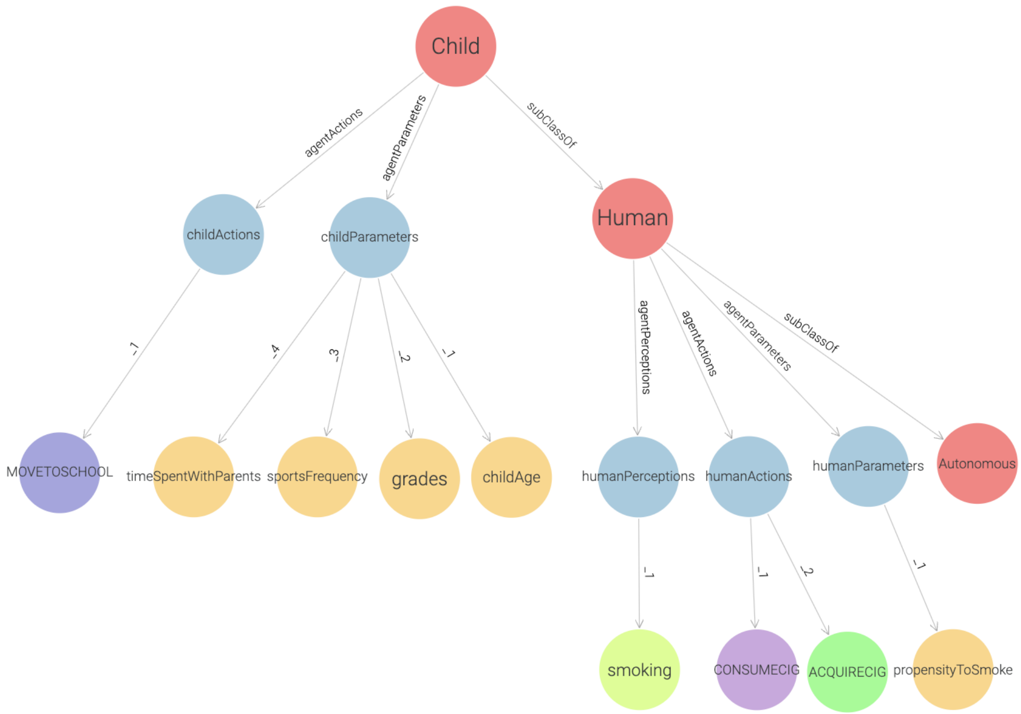

Figure 10: Depiction of how to use the upper ontology terms to describe the *Child* agent in the domain of substance use in the domain-specific ontology, with parameters timeSpentWithParents, sportsFrequency, grades, childAge, and the action of moving to school, and inheriting the parameter propensityToSmoke, the actions CONSUME_CIGARETTE and ACQUIRE_CIG, and the perception of someone smoking from the agent type *Human*.

presence would experience an increase in their parameter *propensity_to_smoke*.

As mentioned in the previous section, actions can possess various attributes, including commitment factor, preconditions, parameter factors, state factors, commitment produced, state produced, and perception produced. While not all actions will make use of these attributes, the following list outlines cases in which the designer may choose to include them:

- Commitment factor: This attribute is relevant when the action is part of an agent's routine, such as the commitment to attend school on weekdays from 8 AM to 3 PM, or when the agent has committed to execute the action at a specific time of day.
- Preconditions: These are necessary conditions for an action to be executed, such as requiring a specific location (e.g., purchasing cigarettes can only be done at a store that sells them), possession of certain resources (e.g., needing a school ID card to enter the school or sufficient quantity of a resource for consumption), or specific

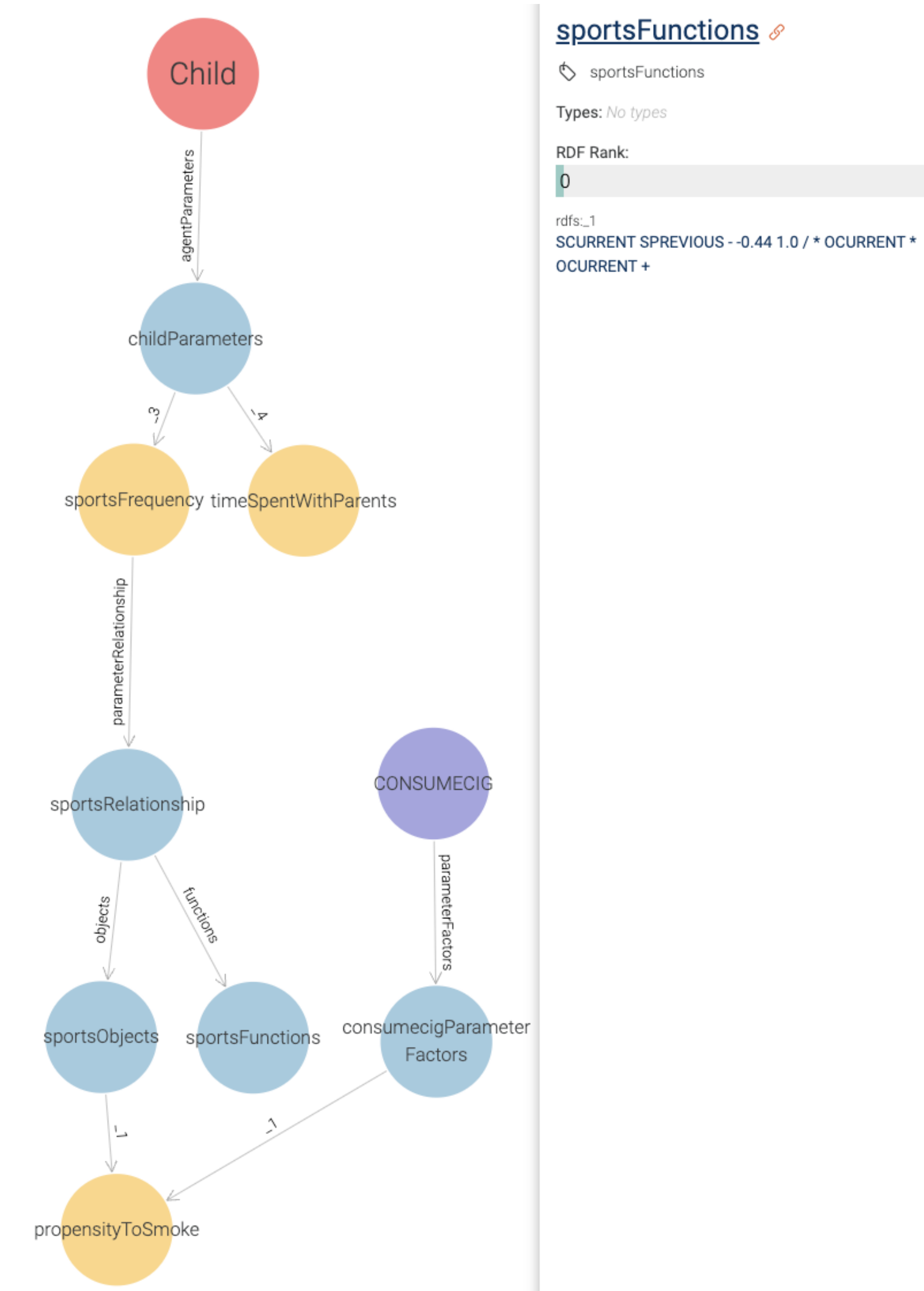

Figure 11: Illustration of how to depict option #1 for the representation of the effects of sports frequency on the propensity to smoke, where propensity is represented as a Parameter, and is in turn affected by the sports frequency, with its corresponding function (sportsFunctions) on the right-hand side in the figure.

Figure 12: Illustration of how to depict option #2 for the representation of the effects of sports frequency on the propensity to smoke, where propensity is represented through the parameter factors sportsFrequency and timeSpentWithParents and their designated parameterWeights (coefficients) in the CONSUME_CIGARETTE action.

times of day (e.g., stores being open only during certain hours).

- Parameter factors: The decision to perform an action is affected by certain parameters, as demonstrated in empirical studies, protective factors against daily smoking include sports frequency, time spent with parents, parental monitoring, and parental support (Kristjansson et al., 2021).

- State factors: The influence of certain memory states on decision-making. For instance, an agent's memory of a recent argument with their parents may affect subsequent decisions.
- Commitment produced: This attribute comes into play when executing an action generates a commitment, such as accepting a friend's invitation to their house, which produces a commitment to carry out the action of visiting at the proposed time.
- State produced: This attribute is applicable when an action results in a new state for the executing agent, such as consuming a specific amount of alcohol leading to the state of inebriation.
- Perception produced: This attribute is relevant when executing an action generates a perception or message for other agents that has a significant and represented impact in the ontology, such as a perception relationship to a parameter.

The action components, in the list above, collectively contribute to the computation of the final action value for each available action in the decision-making step, which is executed every time the agent makes a choice. Consequently, the agent selects the action with the highest value or opts for inaction if the available actions yield negative values. This approach ensures that the agent's decisions are driven by the most favorable outcomes based on the current context and available information.

6 Evaluation

In accordance with the evaluative criteria delineated and the requirements proposed in Chapter 5, the assessment of results includes five key dimensions: *adaptability, extensibility, accessibility, implementation of scenarios,* and *scalability*. As a proof-of-concept, the prototype can only support a certain limited level of evaluation on all of these dimensions, and most of them have only been evaluated to a first minimal level. The exception is the dimension of scenario implementation, which is detailed in Section 6.4, and scalability, in Section 6.6.

6.1 Adaptability

The robustness of the implementation is exhibited in its capacity to accommodate modifications pertaining to both numerical and conceptual representations of a target phenomenon's components. As described in Section 5.4, this adaptability is manifested through the multifaceted ways in which domain experts are enabled to represent concepts and their interactions. Furthermore, the use of Turtle syntax facilitates the ease with which the component values can be altered.

6.2 Extensibility

In its current form, the approach permits the incorporation of additional components into the domain-specific ontology and the upper-ontology. This first can be achieved by aligning them with the pre-established upper-ontology general terms, and the second by adding additional components and changing the structure of the upper-ontology. While there is no cap on the number of components that domain-experts may input, the limitation arises in the unique ways in which the target phenomenon can be represented within the ABM framework.

6.3 Accessibility

A heightened degree of accessibility is conferred by the segregation of domain-specific content from the ABM implementation. Contrary to contemporary implementations that necessitate adjustments to the codebase for alterations in the representation of domain knowledge, the proposed approach and implementation provides a mechanism for these adjustments to be realized without directly manipulating the codebase in the ABM framework. This is facilitated by the provision for designers to modify domain knowledge through changes in the ontologies.

6.4 Scenario Implementation

The scenarios delineated in Chapter 5 were examined with respect to their representation utilizing the upper-ontology concepts and subsequent execution within the simulation framework:

1. **Substance consumption.** Utilizing concepts from the upper ontology, substance consumption can be represented as an action of type *Consume* (resource management). Moreover, multiple *Consume* actions can be specified for the same resource or substance, with variations in the quantity consumed. These variations and the choice of which action to implement, may depend on specific factors, and the outcomes of consuming different amounts may yield increased or distinct effects.
2. **Routine activities.** These can be represented by a commitment and its associated action. In the current implementation, commitments introduce a modifier to the action value associated with an upcoming commitment, enhancing the likelihood of the action's execution when its designated time arrives (through its commitment's time period). Additionally, while engaged in a commitment, other movement actions receive a negative modifier, as they would interfere with the completion of the ongoing commitment. Although negative modifiers are applied, other actions may still be considered if their action components increase their propensity, more so than the negative modifier decreases it.
3. **Protective and risk factors of varying importance and effects.** Numerous methods are available for representing factors using the upper ontology terms. One approach is through their direct impact on other parameters via x-parameter relationships (where x can be parameter, perception, or action), where the nature of the effect is portrayed in the associated negative or positive function. Another method

involves their influence on the propensity to execute an action through parameter or state factors, affecting the final action value with a positive or negative coefficient.

4. **Effects of peer activities on agent behavior.** Peer activities are perceived through perception items and can impact agent parameters specified in the domain-specific ontology via perception-parameter relationships, consequently affecting the behavior or chosen actions for that agent.
5. **Planning.** Planning is accomplished through existing preconditions. Although preconditions associated with an action may not be fulfilled at the time of decision, if the action value for that action is the highest among all available options, the agent will attempt to satisfy the existing preconditions in the following order: resource acquisition, movement, and time. While fulfilling preconditions, the plan to implement the final action may be disrupted if any event altering the parameters or other components associated with the final action value occurs, such as an agent receiving an invitation to play basketball with friends while en route to purchase cigarettes.

All in all, the hours spent on this implementation of a model of peer pressure in the present state of the system counted around 30 hours total for the social science experts and 10 hours for the programmers. It is estimated that about 4 hours of that total was spent on software bugs. For the social scientists, preparation time was around 4 hours, meetings with programmers was 5 hours, and 3 hours was spent on experimentation and reworking the formulas.

Given that the system is still in beta, these are very encouraging numbers, especially considering that no graphical user interfaces were available for the work and the domain experts are not programmers and had never even seen the system before. These results give much reason to continue along the same path.

6.5 Usability of the Present Prototype

There are three main end-user categories for the ultimate version of the *SocialInsight* framework. The largest group are policy makers, the second largest are scientists, with three subgroups (social scientists, psychologists and biologists); the smallest group of end-users is ontology and simulation experts, who expand the system from the perspective of general use across the other two user groups. The present implementation most directly allows for an evaluation with one of these groups, namely, social scientists.

To test the current implementation for end users in the sociology scientist group, an initial development of a model of peer pressure was undertaken. The analysis and usability evaluation was performed by a small group of developers and two domain experts. We took a breadth-first approach, testing both the use of the ontologies, implementation of theory, running the system, and iterating over the design.

The peer pressure model spanned two levels of detail, social and psychological, with a strong emphasis on the former, and involved the creation and experimentation of several simulation agents. Our report on this initial evaluation is detailed below, providing an illustrative example of its present capabilities and areas for enhancement.

1. **Initial Learning Curve.** The effort required to begin using the SocialInsight ABM effectively is comparable to learning any specialized research methodology. Users should ideally have a solid grounding in the theoretical principles underlying their research, along with a basic understanding of ABM operations as tools for scientific inquiry. The introduction of ontological structures within SocialInsight adds a layer of complexity, requiring additional effort to master both practically and methodologically. Supporting documentation is provided to ease this learning process, but users should expect a learning curve similar to that of e.g., mastering multiple linear regression—understanding the methodology first, then its application, over the span of a few weeks of study.
2. **Integration & Experimentation.** The framework allows for relatively straightforward manipulation of parameters and variables, making the creation of new experimental conditions based on emerging hypotheses easier. The logical structure of ontologies and the sub-model of peer influence is well-documented, facilitating the design of experiments and navigation between different layers of the ABM.
3. **Interface Complexity & Technical Support.** The current user interface can be intimidating for non-technical users, especially those without prior coding experience. The system is still in a Proof of Concept (PoC) phase, which makes undertaking complex research projects challenging without technical support. Enhanced user guidance through example documents and clearer explanations of the coding requirements would improve accessibility and usability. On our agenda for the next version of the framework are a more intuitive setup process that includes visual examples that clarify syntax and interaction between model components.
4. **Output Management & Analysis.** Although the transparency of agent actions and decision-making processes in the outputs is a strong point, the current output format makes quantitative analysis and interpretation cumbersome. Improvements in the output interface, such as including plotting and other visual representations of data will significantly enhance the ability to verify model accuracy and interpret results, and to do so efficiently.
5. **Documentation and Transparency Needs.** Greater transparency in the ontology design is necessary in general, and particularly concerning the default mathematical formulas deployed. Detailed and specific explanations of all methodological components within SocialInsight, such as the layering of ontologies and their operational dynamics, are crucial for users to fully leverage the framework's capabilities. This is something that will be addressed in future versions with a graphical user interface.

6.6 Scalability

The existing ABM framework demonstrates proficiency in executing simulations, as outlined in the aforementioned scenarios and in the domain-specific exemplars presented in Section 5.4. The execution time complexity is relatively low, as illustrated in Table 1, where executing a simulation with 10k agents for a total of 93 days (approximately 3 months) takes 68.166 seconds to run. In the context of the substance use and other phenomenon in youth, domain experts may choose to examine the behaviors and outcomes of considerably larger populations, for example with 80k interacting children.

Nr. Agents	Iterations (days)	Total Loop Iterations	Execution Time (ms)
1000	10	170	3041
1000	31	527	4254
10000	31	527	25238
10000	93	1581	68166
80000	93	1581	635590

Table 1: Execution time in milliseconds taken for an increasing amount of agents and iterations (days), where the total number of loop iterations is the number of iterations multiplied by the total amount of hours per day looped by, which is currently 17 hours.

Our approach not only supports large-scale simulations in its current form but also provides a solid foundation for future optimization and scalability. Scalability provisions include for instance extensions to concurrency, where non-neighboring agents can engage in parallel interactions with their surrounding environment and other agents. Additionally, overlay mechanisms may be introduced to reduce time complexity.

7 Conclusions & Future Work

The SIMLife project, and the current implementation of the *SocialInsight* framework prototype, are in an early phase. The methodology proposed encompasses initial ontological structures for representing agents, their parameters, perceptions, available actions, and their subsequent interplay and impact on agent behavior. Although a variety of general conceptual representations for social systems have been proposed within the context of ABMs in prior related work, our approach aims to model a variety of social systems, while enabling the re-usability of existing mechanisms and components across distinct phenomena. In order to achieve these objectives, a list of requirements was formulated for the system, from which it was concluded that the system required a high degree of modularity, not only in the ABM implementation but also in its ancillary components, extensibility accommodating modifications to the model, accessibility, and ability to scale the simulation environment without interfering with domain knowledge provided.

Our current approach has been achieved through a single iteration of the cycle delineated in Chapter 5. To optimize the outcome and improve the approach as a whole, additional iterations will be conducted. Future work includes implementing the approach with more comprehensive and complete scenarios, verifying the ontologies further, as well as facilitating their extension by domain experts in the face of new research. As we develop a more complete framework, we expect to raise the degree of accuracy of its results upon comparison with findings from empirical studies. Finally, in the interest of facilitating the collaboration of a growing variety of domain experts in both the usage of the system and extension of the foundational knowledge base, the development of several graphical user interfaces is crucial. Paired with the modular nature of our approach, we are confident that optimizing accessibility at many levels by means of a graphical user interface, tailored to the research and usability needs of domain experts, will yield significant improvements across the framework.

Given that the current implementation is still in beta, these are very encouraging results, giving much reason to be optimistic and continue this work along the same path, to fulfill the complete vision for the project.

References

Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. P. (2017). Agent based modelling and simulation tools: a review of the state-of-art software. *Computer Science Review*, *24*, 13–33. https://doi.org/doi.org/10.1016/j.cosrev.2017.03.001

Balci, O. (1998). Verification, validation, and testing. *Handbook of simulation*, *10*(8), 335–393.

Baumann, D., Solowjow, F., Johansson, K., & Trimpe, S. (2020). Identifying causal structure in dynamical systems. *Transactions on Machine Learning Research*. https://doi.org/doi.org/10.48550/arXiv.2006.03906

Beheshti, R., & Sukthankar, G. (2014). A normative agent-based model for predicting smoking cessation trends. *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 557–564.

Bobashev, G., Goree, S., Frank, J., & Zule, W. (2018). Pain town, an agent-based model of opioid use trajectories in a small community. *Social, Cultural, and Behavioral Modeling: 11th International Conference, SBP-BRiMS 2018, Washington, DC, USA, July 10-13, 2018, Proceedings 11*, 274–285.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, *99*(suppl_3), 7280–7287.

Borgo, S., Ferrario, R., Gangemi, A., Guarino, N., Masolo, C., Porello, D., Sanfilippo, E. M., & Vieu, L. (2022). Dolce: A descriptive ontology for linguistic and cognitive engineering. *Applied ontology*, *17*(1), 45–69.

Brown, S. H., Elkin, P. L., Husser, C., Bauer, B. A., Lincoln, M. J., Carter, J., Erlbaum, M., Tuttle, M. S., et al. (2004). Va national drug file reference terminology: A cross-institutional content coverage study. *MEDINFO 2004*, 477–481.

Castillo-Carniglia, A., Pear, V. A., Tracy, M., Keyes, K. M., & Cerdá, M. (2019). Limiting alcohol outlet density to prevent alcohol use and violence: Estimating policy interventions through agent-based modeling. *American journal of epidemiology*, *188*(4), 694–702.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems and their applications*, *14*(1), 20–26.

Chen, L., Zeng, W.-M., Cai, Y.-D., Feng, K.-Y., & Chou, K.-C. (2012). Predicting anatomical therapeutic chemical (atc) classification of drugs by integrating chemical-chemical interactions and similarities. *PloS one*, *7*(4), e35254.

Christley, S., Xiang, X., & Madey, G. (2004). An ontology for agent-based modeling and simulation. *Proceedings of the agent 2004 conference*.

da Silva, S. B. V. (2023). Ontology-driven agent-based modeling approach. *MSc Thesis, Dept. Comp. Sci., Reykjavik University*.

Davidovsky, M., Ermolayev, V., & Tolok, V. (2012). A survey on agent-based ontology alignment. *ICAART (2)*, 355–361.

Dray, A., Mazerolle, L., Perez, P., & Ritter, A. (2008). Policing australia's 'heroin drought': Using an agent-based model to simulate alternative outcomes. *Journal of Experimental Criminology*, *4*, 267–287.

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2006). *Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web*. Springer Science & Business Media.

Gotts, N. M., van Voorn, G. A., Polhill, J. G., de Jong, E., Edmonds, B., Hofstede, G. J., & Meyer, R. (2019). Agent-based modelling of socio-ecological systems: Models, projects and ontologies. *Ecological Complexity*, *40*, 100728.

Halpern, J. Y., & Pearl, J. (2013). Causes and explanations: A structural-model approach — part 1: Causes. *CoRR*, *abs/1301.2275*. http://arxiv.org/abs/1301.2275

Hanna, J., Joseph, E., Brochhausen, M., & Hogan, W. R. (2013). Building a drug ontology based on rxnorm and other sources. *Journal of biomedical semantics*, *4*, 1–9.

Kristjansson, A. L., Lilly, C. L., Thorisdottir, I. E., Allegrante, J. P., Mann, M. J., Sigfusson, J., Soriano, H. E., & Sigfusdottir, I. D. (2021). Testing risk and protective factor assumptions in the icelandic model of adolescent substance use prevention. *Health education research*, *36*(3), 309–318.

Kristjansson, A. L., James, J. E., Allegrante, J. P., Sigfusdottir, I. D., & Helgason, A. R. (2010). Adolescent substance use, parental monitoring, and leisure-time activities: 12-year outcomes of primary prevention in iceland. *Preventive medicine*, *51*(2), 168–171.

Kristjansson, A. L., Sigfusdottir, I. D., & Allegrante, J. P. (2013). Adolescent substance use and peer use: A multilevel analysis of cross-sectional population data. *Substance abuse treatment, prevention, and policy*, *8*, 1–10.

Lamy, F., Perez, P., Ritter, A., & Livingston, M. (2011). An agent-based model of alcohol use and abuse: Simarc.

Lamy, F., Bossomaier, T., & Perez, P. (2015). An ontologic agent-based model of recreational polydrug use: Simuse. *International Journal of Simulation and Process Modelling*, *10*(3), 207–222.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association*, *88*(3), 265.

Livet, P., Phan, D., & Sanders, L. (2008). Why do we need ontology for agent-based models? *In Complexity and artificial markets*, *13*, 133–145. https://doi.org/doi.org/10.1007/978-3-540-70556-7_11

Lovrencic, S., & Cubrilo, M. (2008). Ontology evaluation-comprising verification and validation. *Central European Conference on Information and Intelligent Systems*, 1.

Luke, S., Cioffi-Revilla, C., Panait, L., & Sullivan, K. (2004). Mason: A new multi-agent simulation toolkit. *Proceedings of the 2004 swarmfest workshop*, *8*(2), 316–327.

McBride, B. (2001). Jena: Implementing the RDF model and syntax specification. *SemWeb*, *1*, 23–28.

Miller, E. (1998). An introduction to the resource description framework. *D-lib Magazine*.

Minar, N., Burkhart, R., Langton, C., Askenazi, M., et al. (1996). The swarm simulation system: A toolkit for building multi-agent simulations.

Peters, J., Bauser, S., & Pfister, N. (2020). Causal models for dynamic systems. *Probabilistic and Causal Inference: The Works of Judea Pearl*, 671–690. https://doi.org/doi.org/10.48550/arXiv.2001.06208

Phan, D., Livet, P., Müller, J.-P., & Sanders, L. (2010). Ontology, a mediator for agent-based modeling in social science. *Journal of Artificial Societies Social Simulation*, *13*, 13–33. https://doi.org/10.18564/jasss.1538

Railsback, S. F., & Grimm, V. (2019). *Agent-based and individual-based modeling: A practical introduction*. Princeton university press.

Schwartz, S. (2006). A theory of cultural value orientations: Explication and applications. *Comparative sociology*, *5*(2-3), 137–182.

Sigfusdottir, I. D., Farkas, G., & Silver, E. (2004). The role of depressed mood and anger in the relationship between family conflict and delinquent behavior. *Journal of youth and adolescence*, *33*(6), 509.

Sigfúsdóttir, I. D., Thorlindsson, T., Kristjánsson, Á. L., Roe, K. M., & Allegrante, J. P. (2009). Substance use prevention for adolescents: The icelandic model. *Health Promotion International*, *24*(1), 16–25.

Stankov, I., Yang, Y., Langellier, B. A., Purtle, J., Nelson, K. L., & Diez Roux, A. V. (2019). Depression and alcohol misuse among older adults: Exploring mechanisms and policy impacts using agent-based modelling. *Social psychiatry and psychiatric epidemiology*, *54*, 1243–1253.

Thorisdottir, I. E., Sigurvinsdottir, R., Kristjansson, A. L., Allegrante, J. P., Lilly, C. L., & Sigfusdottir, I. D. (2020). Longitudinal association between social media use and psychological distress among adolescents. *Prev Med*, *141*, 106270.

Thorisdottir, I. E., Agustsson, G., Oskarsdottir, S. Y., Kristjansson, A. L., Asgeirsdottir, B. B., Sigfusdottir, I. D., Valdimarsdottir, H. B., Allegrante, J. P., & Halldorsdottir, T. (2023). Effect of the covid-19 pandemic on adolescent mental health and substance use up to march, 2022, in iceland: A repeated, cross-sectional, population-based study. *The Lancet Child & Adolescent Health*, *7*(5), 347–357.

Thórisson, K. R., Saemundsson, R. J., Jonsdottir, G. R., Reynisson, B., Pedica, C., Thrainsson, P. R., & Skowronski, P. (2009). Applying constructionist design methodology to agent-based simulation systems. *3rd International KES Symposium on Agents and Multi-agent Systems – Technologies and Applications*.

Thórisson, K. R., Spivack, N., & Wissner, J. M. (2010). The semantic web: From representation to realization. *Transactions on Computational Collective Intelligence II, Lecture Notes in Computer Science*, *6450*, 90–107.

Thórisson, K. R., Bieger, J., Thorarensen, T., Sigurdardottir, J., & Steunebrink, B. (2016). Why artificial intelligence needs a task theory – and what it might look like. *9782*, 118–128. https://doi.org/10.1007/978-3-319-41649-6_12

Tisue, S., & Wilensky, U. (2004). Netlogo: Design and implementation of a multi-agent modeling environment. *Proceedings of agent*, *2004*, 7–9.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, *11*(2), 93–136.