RESEARCH ARTICLE

An Ontological Analysis and Assessment of Human Genome Conceptual Models

Alberto Garcia S.[1*], Anna Bernasconi[1,2], Giancarlo Guizzardi[3,4], Oscar Pastor[1], Veda C. Storey[5] and Ignacio Panach[1]

*Correspondence:
algarsi3@pros.upv.es
[1]PROS Research Center & VRAIN Research Institute, Universidad Politècnica de València, Camino de Vera S/N, 46021 Valencia, Spain
Full list of author information is available at the end of the article

Abstract

Background: The ability to sequence the human genome is a scientific, historical breakthrough. The human genome mapping is available to all scientists, but information about it can be difficult to share. The previously developed Conceptual Schema of the Human Genome represents the relevant concepts required to holistically understand the human genome by studying the human genome from a global perspective without focusing on a specific dimension.

Results: In this paper, we present our efforts to ensure that the human genome concepts can be meaningfully shared, by conducting an *ontological unpacking* to facilitate domain understanding and data exchange among heterogeneous systems. The ontological unpacking is an analysis based on a foundational ontology that enriches the input conceptual model, The analysis and enrichment process are supported by the ontology-driven conceptual modeling language, OntoUML, which has previously been applied to complex models to gain ontological clarity. The results lead to major, diverse modeling implications, including the: i) characterization of biological entities; ii) changes in biological entities over time; and iii) representation of chemical compounds. The value of this method is demonstrated by an empirical evaluation that captures the differences that occur when exploring a new domain by adopting a traditional conceptual model and comparing it to its related ontologically unpacked model.

Conclusion: Our research is evidence that including a strong ontological foundation in traditional conceptual models is useful. It contributes to designing models that capture the particularities of biological domains better than the original models. This evidence is corroborated by the statistically significant results of an empirical study that evaluates how the use of an ontological conceptual modeling language is perceived.

Keywords: Ontological Unpacking; Conceptual Modeling; Foundational Ontology; OntoUML; Genomics; Metabolic Pathways; Data Integration

Background

The modeling of the human genome is a fascinating and extremely important area of research due to its potential to impact all of mankind through improved treatments and possibly, removal of diseases. In essence, this modeling contributes to understanding life itself. Progressing research on the human genome, however, is challenged for many reasons, perhaps the greatest of which is the fact that the body of knowledge surrounding the human genome constantly changes and evolves as scientists and researchers all over the world conduct research with it. Furthermore, the terminology and concepts employed in genomics can be imprecise and

continuously changing, as are the scope and complexity of the modeling required to represent them. The definitions of terms needed to characterize any phenomena rely on the experience of the domain experts who use and interpret them. Definitions may be purposely abstract to reflect the constantly changing knowledge of a domain. However, these terms cannot simply be translated into an unambiguous representation of that knowledge. Consider the term allele that might refer to: i) an alternative form of a gene or locus[1]; ii) one of two or more possible forms (i.e., the specific DNA sequence) of a particular gene; or iii) one of a set of coexisting sequence alleles of a gene. These definitions, however, are imprecise. Does an allele describe a specific change on a specific sequence or a more general change on an undefined sequence? Even worse, the term allele is also used to describe changes in DNA sequences of regions not associated with any gene. How can this concept (and its multiple underlying interpretations) be represented in a consistent way? How can we create knowledge from such concepts? A fundamental prerequisite for analyzing and understanding any complex domain, is to facilitate a shared understanding among the people who work in that domain.

The most common artifacts used for representing concepts in a consistent way and for facilitating a shared understanding of genomics are so-called lightweight ontologies (i.e., logical specifications typically in some form of Description Logics) and thesauruses of controlled vocabulary [1], because they provide standard concepts and definitions. These lightweight ontologies favor agility in contrast to having formal and ontological coherence [1]. They are also limited in that they can only correctly represent a minor portion of relevant facts in genomics [2]. Representing probabilistic knowledge using these ontologies tends to produce erroneous models [3]. Therefore, a complementary approach is needed. Conceptual models are appropriate because they facilitate the exchange of information [4, 5, 6], while providing a sound basis to make a particular conceptualization process explicit and facilitating the achievement of a shared understanding of a domain [7]. For the human genome domain, applying conceptual modeling can: improve communication among physicians, geneticists, biologists, and other researchers; assist in knowledge transfer; and, ultimately, enable efficient exploitation of information for progressing the understanding of the human genome [8].

Prior research has created a Conceptual Schema of the Human Genome (CSHG) [9]. The objective of this research is to extend this conceptual model by making the definition of the relevant concepts of the model precise, explicit and understandable for all. To do so, we conduct an ontological analysis and enrichment of the current model. We use the adjective "ontological" in a strong sense in that our analysis aims at revealing and explicitly modeling a number of aspects related to the *nature* and *real-world semantics* of entity types and relationships in this domain. We employ the conceptual modeling language OntoUML [10], which is grounded in the Unified Foundational Ontology (UFO) [11]. Our first contribution is to reformulate the conceptual model, showing how a foundational ontology brings ontological clarity to complex models by facilitating domain understanding and data exchange among heterogeneous systems [12]. Our second contribution is to assess the value of the

[1]A locus is a specific region of a chromosome that can contain a gene or another sequence of interest.

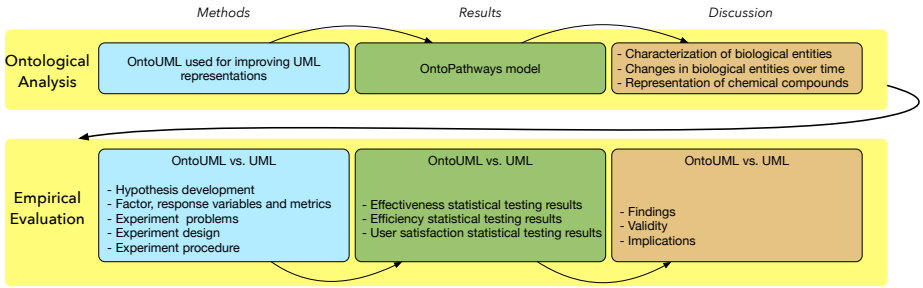


Figure 1: Overview of research.

ontological unpacking process by performing an empirical study. The results of that study show that OntoUML representations are more effective in explaining the observed domain with respect to UML representations. This unpacking is expected to produce models that capture and describe the particularities of a domain better than the original model.

Paper organization. In the remainder of this section, we review related research, introduce important concepts needed to understand the remainder of our work, and introduce the Conceptual Schema of the Human Genome (CSHG), which we subject to our analysis and redesign. Our core contribution, as shown in Figure 1, is partitioned into two parts (as shown in the Methods, Results, and Discussion sections).

- *Ontological analysis.* Using OntoUML (Methods) results in an ontologically grounded model of the CSHG (Results) that has several implications for different biological understanding aspects (Discussion).
- *Empirical evaluation.* The OntoUML model is compared to its corresponding (original) UML model to understand if it better serves the purpose of explaining a complex domain. For example, for a specific model of metabolic pathways we build a study with hypothesis and experimental design (Methods). We measure and compare effectiveness, efficiency, and user satisfaction with the two methods (Results), and finally elaborate our findings discussing their validity and implications (Discussion).

Related work

Previous work on the ontological unpacking of biology-related models has been performed in [12], where the method has been applied to the case of a Viral Conceptual Model [13] designed to organize the data collected about SARS-CoV-2, the virus responsible for COVID-19, as well as similar viruses. In [14] we framed our first proposal to use ontological unpacking in a conceptual model of the human genome, which is further developed and detailed here, by exploring the pros and cons of the related efforts. Ontology driven conceptual modeling has been compared to traditional conceptual modeling in [15], followed by other studies that have considered their differences [16, 17] in various domains or from a theoretical point of view [18]. Here, we do not compare different languages or paradigms, but rather, we compare the capability of different models (an original conceptual model and its ontologically

unpacked version) to completely and unambiguously represent a domain. Doing so, serves the intended purpose of explaining that domain to a non-expert user that approaches it for the first time.

Background concepts

Figure 2 guides the presentation of the context upon which this research is based. Traditional conceptual modeling [19] was conceived for representing artifacts and their semantics, associated with databases or software. It is generally described as the activity of representing aspects or artifacts of the physical and social world with a descriptive or communicative purpose [20]. A conceptual model is a representation of a system that consists of a set of concepts used to help people know, understand, communicate, or simulate a subject that the model represents. In contrast, ontology-based conceptual modeling derives from the use of ontological theories (conceived by the formal ontology, cognitive science and philosophical logic-related fields), to develop engineering artifacts (e.g. modeling languages, methodologies, design patterns and simulators) that improve the practice of conceptual modeling [21]. The purpose of the two kinds of modeling are different: the first aims to describe conceptualizations while the second pursues their explanation. Explanation can be achieved by grounding the modeling exercise on a Foundational Ontology, such as, the UFO [10], as used in this paper, the Basic Foundational Ontology (BFO, [22]), the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE [23]), or the Business Object Reference Ontology (BORO [24]). In contrast, traditional conceptual models are not driven by any meta-model or ontological foundation.

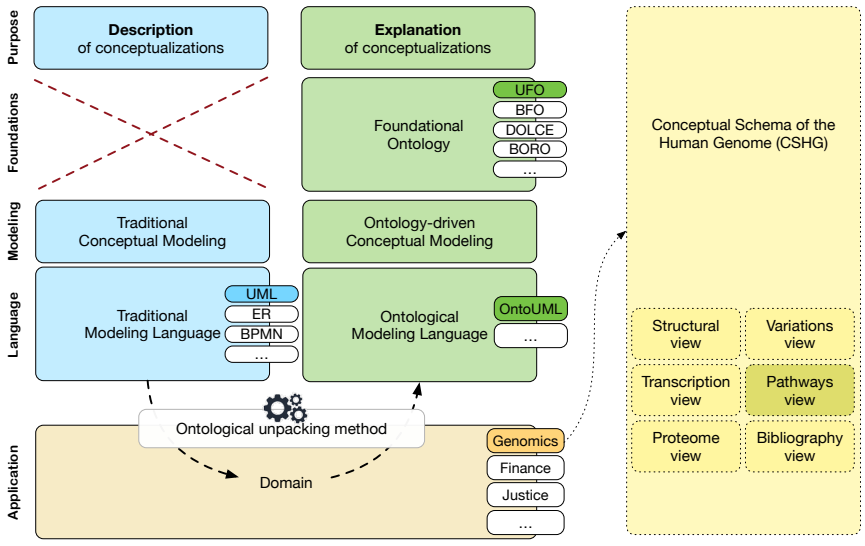


Figure 2: Schematic representation of concepts used in this manuscript and the context within which the 'ontological unpacking' method is performed.

Several languages enable us to pursue the modeling effort, among which UML [25], ER [19], and BPMNs [26] are well known. Ontological Modeling requires more

expressive languages. OntoUML is an ontology model language that is grounded on the UFO ontology. In OntoUML, classes are enriched with additional semantics by means of stereotypes. Few alternatives are found in the literature (e.g., [27]).

In the following, we describe our approach to ontological unpacking, where we use OntoUML (based on UFO) on the complex domain of genomics and test our method on a specific portion (the pathway view) of a conceptual model of the human genome. The same domain fragment has been first represented using a traditional modeling language, i.e., UML. Ontological unpacking refers to the process of revealing relevant knowledge that remained implicit by means of transforming a UML model into its corresponding OntoUML version.

Conceptual Schema of the Genome

Prior research on modeling the human genome resulted in the development of the Conceptual Schema of the Human Genome (CSHG). Since our understanding of genomics evolves rapidly, so does the CSGH evolve. The initial conceptual model focused on representing the most relevant concepts when studying genomics, such as chromosomes, genes or variations and basic participants in the transcriptome and proteome steps [28]. This model was expanded to include the concept of phenotype and its relationships with other genomics components [29]. The second version drastically changed how the DNA sequence is represented: from a gene-centric to a chromosome-centric vision [30]. This version included the chromosome element class, for an increased generalization of the elements that can be identified in a DNA sequence. Any sequence with a specific functionality can be characterized (e.g., enhancers, promoters). The third version expands the representation of the transcription process; re-evaluated the characterization of variants; included changes caused by variations at the DNA, RNA and amino acid levels; and increased the generality of multiple concepts.

Creating a holistic Conceptual Schema of the Human Genome requires integrating conceptual components that represent the relevant data that connect the genome structure (genotype) with its expression of real world behavior (phenotype). The evolution of the schema resulted in different views (components):

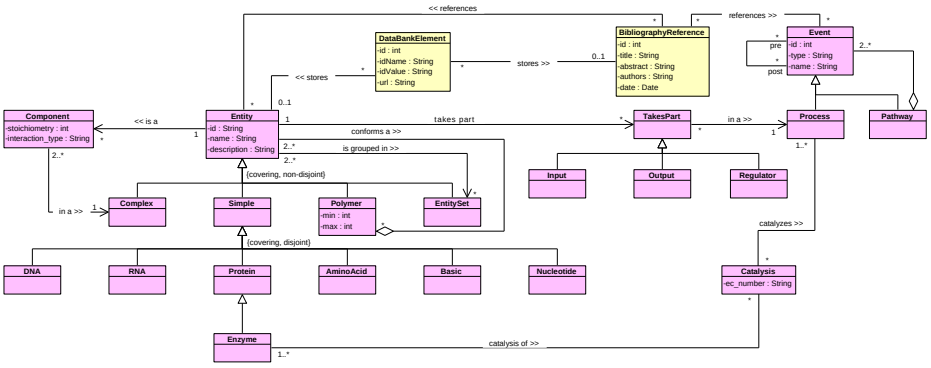
- Structural view, which focuses on the composition of transcribable chromosome elements (genes, exons, regulatory elements, conserved regions, etc.).
- Variations view, which identifies the types of changes that may occur in the genome.
- Transcription view, which deals with the process of moving from DNA to RNAs.
- Pathways view, which describes the chemical reactions that explain the different molecular processes.
- Proteome view, which characterizes proteins structure and properties.
- Bibliography and data sources view, which identifies relevant information related to sources of valid information (publications, genome data sources...).

These views have many practical uses, such as: identifying and managing genomic variations related to the treatment of Alzheimer's [31]; developing a conceptual model-based framework to improve the data quality processes of precision medicine [32]; reporting early diagnosis of alcohol sensitivity [33]; identifying variations that play a role in developing colorectal cancer [34], improving the diagnosis

of congenital cataracts [35]; improving the prioritization of genetic variations [36]; and increasing interaction and collaboration in diagnosing genetic diseases [37].

Here, we focus only on the Pathways view because it reflects very critical aspects of the genomics domain, including a *biological event* that addresses how genome elements interact to produce a biological behavior. Given its importance and richness, this view provides an appropriate way to motivate and demonstrate the need for the type of analysis and redesign proposed here. The Pathways view is depicted in Figure 3 as a UML class diagram; that is, using the standard expressivity provided by UML. The model is centered on the notions of entity and event, represented by the homonymous classes.

Figure 3: **Pathway view of the CSHG.** Pink boxes represent the classes that pertain to the Pathway view. Classes in yellow pertain to the bibliography and data sources view.



Consistent with the term used by geneticists, an `Entity` class identifies any possible physical component present in our body and plays a role in an event. In turn, an `Event` class represents the biological events that occur in a body. Events are recursively composed of additional events. An event can be a `Pathway` (complex event, made up of other events) or a `Process` (elementary event). A process is then an atomic, simple event of a given type. A pathway is a more complex type of event that is decomposed into a specific set of events, either processes or pathways. Entities are of different types: `Simple`, `Complex`, `EntitySet`, or `Polymer`. A simple entity represents the elementary biological physical components that interact in processes such as proteins (e.g., the Breast cancer type 2 susceptibility protein). They appear in the schema with the `Protein_e`[2] class, or as chemical compounds (e.g., water), represented with the `Basic_e` class.

Simple entities can be combined to form entities that are complex entities, entity sets, or polymers. A complex entity is created when at least two entities are combined to create a new one. Each entity plays a specific role in the complex (e.g., has a specific type of interaction and a stoichiometric ratio). This is represented through the `Component` class that associates a given entity with a complex entity. An entity set represents a number of biological entities of any type that can be

[2]In [9], all "Simple" classes have their names appended with "_e".

used interchangeably because they play an equivalent role. Unlike complexes, the entities that belong to an entity set retain their individuality; that is, they play an equivalent role but are not combined. Entity sets are used as aggregates to reduce the granularity of pathways. A polymer is created when an entity is concatenated a specific number of times (at least two). Unlike complexes and entity sets, the polymer is made up of only one type of entity, represented in the conceptual model as an Object Constraint Language (OCL) integrity constraint (identified as IC-1 in the schema).

A process is a specific interaction between entities. An entity can participate as an `Input`, an `Output`, or a `Regulator`. These associated sets of inputs, outputs, and (optionally) regulators characterize the process functionality. Therefore, when an entity takes part in a specific process, it assumes one of these three roles. Another dimension is the `Catalysis`, which is the increase of the reaction rate of a process. The reaction rate is the rate at which a process takes place. Processes are catalyzed by enzymes, a special type of protein.

The `BibliographyReference` class supports the information used to represent pathways. The two main notions of entity and event can be linked to the bibliography references that report relevant information. The `DataBankElement` class enables referencing the external sources where bibliographic references and proteins are stored. This representation provides an effective way to generate a snapshot of current, available knowledge in the scientific community of the internal working mechanisms of the human body.

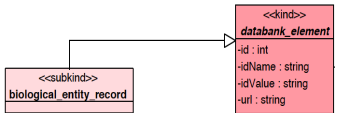
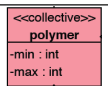
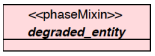
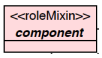
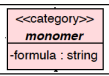
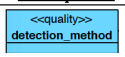
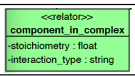
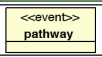
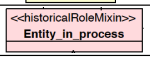
Ontological analysis

Method

OntoUML is an ontology-driven conceptual modeling language based on the upper ontology Unified Foundational Ontology (UFO, [10, 11]). OntoUML uses stereotypes to represent the mapping between its modeling constructs and UFO ontological categories. OntoUML is built upon the fundamental distinction between Types and Individuals. Types are patterns of features that are repeatable across multiple instances. OntoUML includes a theory of higher-order types so first-order types are types instantiated by individuals, whereas higher-order types (represented by the stereotype «type») are instantiated by other types (e.g., the types Emperor Penguin and Golden Eagle are instances of the higher-order type Bird Species). UFO countenances two fundamental types of individuals: endurants (i.e., objects and their existentially dependent reified aspects) and perdurants (i.e., events and processes).

Endurants types are classified based on two dimensions, sortality (identity) and rigidity. Sortals are types whose instances obey a single identity principle (i.e., are all of the same «kind»); non-sortals are types that classify instances of multiple kinds. A type is rigid if it defines essential characteristics of its instances; anti-rigid if it defines contingent characteristics for all instances. The type person is typically considered rigid (since instance of person are necessarily so), but the type student considered anti-rigid (since no student is necessarily a student). Kinds represent the genuine fundamental types of objects that exist according to a particular conceptualization of a domain. All objects belong to exactly one kind. However, there can be other static specializations of a kind, namely «subkinds»; e.g., the kind “gene product” can be specialized into the subkinds “coding RNA” and “non-coding RNA”.

Figure 4: Overview of a part of OntoUML stereotypes, with their description and examples taken from the proposed ontologically unpacked model.

Stereotype	Description	Example
«type»	High-order type whose instances are themselves types.	
«vkind» and «subkind»	1) Type of objects that exist according to a particular conceptualization of the given domain. These fundamental types describe what the objects in that domain essentially are. 2) Subdivision of a kind.	
«collective»	Plural entity that aggregates parts (members), all of which play the same role with respect to the whole.	
«phase» and «phaseMixin»	1) Anti-rigid sortal type that captures a cluster of change conditions in intrinsic properties 2) Anti-rigid non-sortal type that captures a cluster of change conditions in intrinsic properties, for instances of multiple kinds	
«role» and «roleMixin»	1) Relationally dependent universal, capturing relational properties shared by instances of a given kind 2) Role for types that represent properties shared by entities of multiple kinds	
«category»	Necessary properties that are shared by entities of multiple kinds.	
«quality»	Aspect that can be directly associated with structured value spaces.	
«relator»	Truth-maker of relational propositions. Relations (as classes of n-tuples) can be completely derived from relators.	
«event»	Class whose instances are events	
«historicalRole» and «historicalRoleMixin»	1) Role played by sortal objects in an event 2) Role played by non-sortal objects in an event	

Objects can also be classified depending on their principle of unity, i.e., the principle binding the parts that form a whole. For example, they can be «collectives» if they are composed of parts (termed *members*) that play the same role with respect to the whole, or *functional complexes* if they are composed of parts (termed *components*) that play different roles with respect to the whole. Finally, objects can be «quantities» to represent homeomerous entities (i.e., entities repeatably decomposable into entities of the same kind), such as water, sand, or blood. Since most of the kinds in a domain are those whose instances are functional complexes, we use the stereotype «kind» simply to represent them.

Anti-Rigid types are specialized into «phases» and «roles». Both phases and roles are dynamic types. Phases have intrinsic dynamic classification conditions; i.e., they capture a cluster of change conditions in intrinsic properties. Roles, in contrast, have relational dynamic classification conditions; i.e., they capture a cluster of change conditions bound to changes in a relational context. For instance, a blood cell has multiple phases such as blood stem cell, red blood cell, etc. depending on its maturity (an intrinsic property). In the case of roles, a person (an instance of the kind person) can be a patient (role) while participating in a medical treatment.

Phases and roles are sortals (i.e., they classify things of the same kind). We can, however, have analogous anti-rigid non-sortal classes, namely, «phaseMixins» and

«roleMixins». As non-sortals, phaseMixins and roleMixins classify instances of multiple kinds. For instance, suppose a protein (kind) and an organic chemical compound (kind) play the role of a regulator in a specific biological process. There are two different roles: the "regulator protein" and the "regulator chemical compound". Both regulate a process so we can abstract them into a new roleMixin, called regulator, from which the other two roles specialize. PhaseMixins and roleMixins can be thought as refactoring classes (abstracting properties common to entities of multiple kinds) and, hence, they are always *abstract* types (i.e., types that cannot be directly instantiated). We can have refactoring (non-sortal) types that are rigid, i.e., that abstract *essential* properties common to entities of several kinds. These are marked as the «category» stereotype.

Objects bear a number of *aspects*, some of which are intrinsic to them (i.e., existentially depend solely on them). These are termed «qualities» or «modes». Qualities are aspects that can be directly associated with structured value spaces (e.g., color or temperature); modes are full-fledged object-like entities with their own aspects but which are still existentially dependent on some bearer. Besides intrinsic aspects, we have relational ones, i.e., entities that are existentially dependent on a multitude of individuals, thus, binding them. These are termed «relator». Relators are the truth-makers of material relations. For instance, the "participation in trial" relator connects a patient with a clinical trial.

Besides endurants, OntoUML has perdurants to represent events [38]. Events are characterized with the «event» stereotype. They have their own properties and can be decomposed. Events are immutable because they only exist in the past. Endurants and perdurants interact in several ways. For example, endurants *participate* in events, are *created* by events, and are *terminated* by events. Finally, since events as particularized instances that only exist in the past, roles played by objects in an event (i.e., while an event was occurring) are termed «historicalRoles» (or «historicalRoleMixins», depending whether they are sortals). Figure 4 summarize the main message of each stereotype and provides an example taken directly from the ontologically unpacked model described in the results.

Results

We review the original conceptualization underlying the CSHG by means of an ontological analysis mediated by OntoUML and its underlying foundational ontology. The results lead us to an improved CSHG, whose sound and precise ontological commitment fulfills the conceptual clarification our work explores (see Figure 5). This analysis focuses on clarifying the notions of entity and event in the original model and how each of them relates to the other.

In the original UML class diagram of Figure 3, the concepts of `Entity` and `Event` are represented as simple classes. However, their exact conceptual characterization can be made explicit by using OntoUML's finer-grained class and association constructs (reflecting UFO's distinctions among endurant and event types and relations).

The entity concept (`biological_entity` class in the unpacked model) defines a set of very diverse molecules with different identity principles. Therefore, we annotated the concepts of biological entity and `simple` entity with the «category»

stereotype, since categories aggregate essential properties of individuals that follow different identity principles (belong to different kinds).

The entity concept (renamed to `biological entity`) is used to define every physical entity that can have a role in one or more processes. This definition implies that these entities are very diverse and have different identity principles. Therefore, we annotated the concept of entity and the concept of simple with the «category» stereotype because categories aggregate essential properties to individuals that follow different identity principles (belong to different kinds).

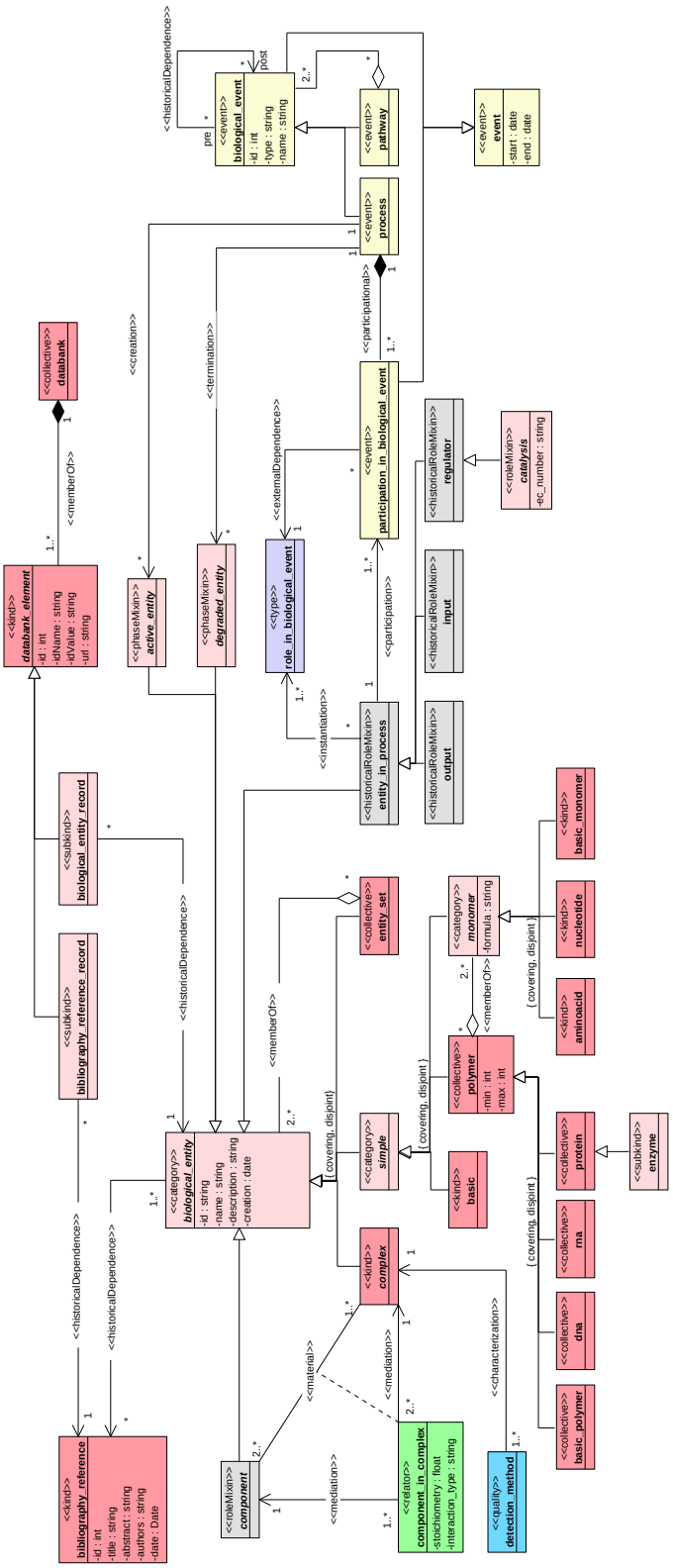
In OntoUML, we added the stereotype «event» to the event concept (renamed to `biological_event` in the unpacked model) to represent that they are ontological entities that unfold over time, accumulating temporal parts and mapping the world from situation to situation [38]. Events are of great importance in human cognition, with the need to model them explicitly. By modeling events as classes, we provide identity principles and properties, as well as rules for relating various event types. The UML version of the model was characterized by an identifier and a name. By mapping our original class named `Event` to its corresponding notion of `event` in OntoUML [38], we add two new attributes, *begin* and *end*. This is because events in OntoUML are framed by specific time intervals. The addition of these temporal attributes supports reasoning with Allen's time interval relations [39], as well as distinguishing, e.g., cases in which an event is eventually followed by another (i.e., after, in Allen terms) from cases in which an event is immediately followed by another (i.e., meet), etc.

In the original model, we had one type-reflexive relation connecting the event class with itself and with the rolenames "-Pre" and "-Post". That modeling choice, however, left ambiguous whether this relation represented a mere temporal precedence between occurrences or a stronger causal connection. To make explicit that the intended semantics referred to the latter, we used OntoUML's «historicalDependence» stereotype [38]. This makes explicit that, if an event of type A is historically dependent on a event of type B, then instances of A must necessarily be preceded by instances of type B. Historical dependence implies temporal precedence, but not vice versa.

For CSHG, following the structural dimension, there are two types of events: the process and the pathway. This dimension is represented through an aggregation relationship with the event class. Following the language's imposed mereological theory, complex entities must be composed of at least two disjoint parts (the *Weak Supplementation Axiom* [40]) with minimum cardinality constraints on the relations. This revised part of the model is a direct instantiation of UFO's structural partonomy pattern [40].

The participation dimension is characterized by representing the role that biological entities play in processes. This was originally modeled by the `Takes_part` class in the UML schema, where we showed that an entity can act as an `Input`, an `Output`, or a `Regulator` in a process. This representation has been expanded in the OntoUML version of the schema. First, we created a set of classes (i.e., `entity_in_process` and its specialized classes) stereotyped with «historicalRoleMixin» to indicate playing roles, which biological entities have participated in, as an event. Differently than in the UML schema, the minimum cardinality of the association between the historical role and the process is one. For a biological entity to play the role, it must

Figure 5: **Ontologically enriched version of the CSHG.**



have mandatorily participated in an event. Historical roles explicitly describe the variety of roles that biological entities may play in the processes.

`biological_events` depend on `biological_entities`. Since atomic events (i.e., `processes`) are directly existentially dependent on `biological_entities`, we can use the extensionality principle of the event mereology to derive the existential dependency of complex processes (i.e., `pathways`). In addition, the defined roleMixin (i.e., `entity_in_process`) allows for creating "portions" to describe the specific participation of an entity. We created the `participation_in_biological_event` class, stereotyped as «event», to divide an event into the individual participation of biological entities. Every instance of this class is derived from parthood and existential dependence, and is bound to a specific subtype of a historical roleMixin (e.g., input, output, regulator, among any other role that can be discovered). Making explicit the notion of participation is of great importance from an ontological point of view. For instance, the process by which proteins are synthesized (translation) can be decomposed into atomic steps (e.g., initiation, elongation, and termination) to model the "constructed" dimension by creating segments using temporal schemes as external references. It can also be decomposed into portions that encapsulate the participation of biological entities in the whole process (e.g., participation of the ribosome and the mRNA strand).

Another capability of the schema, which is enabled by the use of the «event» stereotype, is that we can model the creation and termination of biological entities. Millions of molecules are created and destroyed by different events that occur in our body, which is a special type of participation of endurants (i.e., biological entities) in events. To represent this situation, we modeled two phases to represent whether an entity exists or has been destroyed (i.e., the `active_entity` and `degraded_entity` classes). The «phaseMixin» stereotype is used to represent changes in intrinsic properties of kinds (i.e., if it is destroyed or not). If a biological entity is related to an event using an association stereotyped with «creation», that entity is created in that event. Similarly, for the «termination» stereotype. Besides, we included the *creation* attribute to identify when a biological entity was created.

One goal of applying the ontological analysis was to assess whether some of the modeled concepts in the UML schema were redundant. For instance, do biological entities that are both simple and polymer exist? The answer is yes: proteins are modeled through the `Protein_e` and the `Polymer` classes (since a protein is a polymer of amino acids). This led us to the next question: should proteins (polymers) and amino acids (the atomic elements that compose them) be modeled at the same level of hierarchy (as a type of simple entity)? The answer is no, because one is composed of the other; one is atomic and the other is not. As a result of our exercise, we reduced the number of concepts into which a biological entity can be specialized in the unpacked version: `complex`, `simple`, and `entity_set`.

A `complex` entity is a functional complex that we stereotyped as a «kind». It represents a rigid concept providing an identity principle. A complex entity is created when at least two biological entities are combined. Each of the entities forming a complex, is called a `component`, and plays a specific role within that whole. Therefore, we annotated the component class with the «roleMixin» stereotype. This stereotype represents changes caused by relational contexts (i.e., a biological entity of any types being part of a complex). A «material» relationship between the

component and the complex exists to represent that complexes are made of components that are connected in particular ways. This relationship is materialized with the `component_in_complex` class, annotated with the «relator» stereotype. An instance of `component_in_complex` must exist in order to connect a component and a complex. Since complexes are made of multiple biological entities, at least two instances of `component_in_complex` per complex must exist. This truth-maker of the relation shows the method used to detect the component and how it interacts with the complex.

We stereotyped the `simple` entity as a «category», just like the `biological_entity` class. The `polymer` class has been reevaluated so it is a type of a `simple` element, and new class: the `monomer`. A polymer is stereotyped as a «collective» that is composed of a single type of monomer. The monomer is a «category» that groups the set of different atomic elements that can conform polymers. The monomer is characterized by its chemical formula. There are three types of monomers: the `aminoacid`, which aggregates to create proteins; the `nucleotide`, which aggregates to create DNA and RNA elements; and `basic_monomer`, which clusters other monomers such as glucose. Finally, the `basic` entity remain unchanged as a type of simple entity.

We stereotyped the `entity_set` entity as a «collective» to identify plural entities that aggregate parts (members) that play the same role with respect to the whole. This definition captures perfectly the essence of the entity set because it is a group of multiple biological entities (the parts) that play the same role with respect to a process (the whole). The new characterization of biological entities becomes clearer.

In OntoUML, we stereotyped the `databank` concept as a «collective» whose members are the different records of the database. (i.e., the `databank_element` class that is stereotyped as a «kind»). A new addition is the specialization of the `databank_element` into two new classes (e.g., `biological_entity_record` and `bibliography_reference_record`) stereotyped as «subkind». The first subkind refers to records of biological elements; the second to records of bibliographic references.

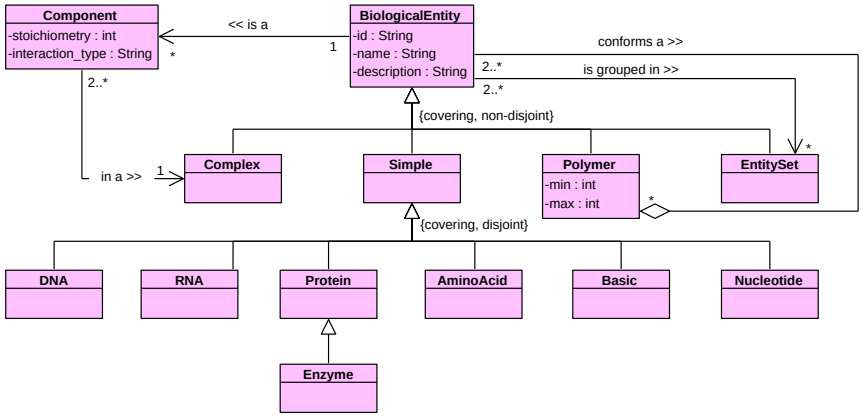
Discussion

The ontological analysis was able to identify, reveal, and propose changes to several aspects of a model (created in the traditional way) in order to better grasp the domain semantics. The benefits of doing so can be measured in terms of sub-parts of the model. The main implications can be summarized in three areas: characterization of biological entities; changes in biological entities over time; and representation of chemical compounds.

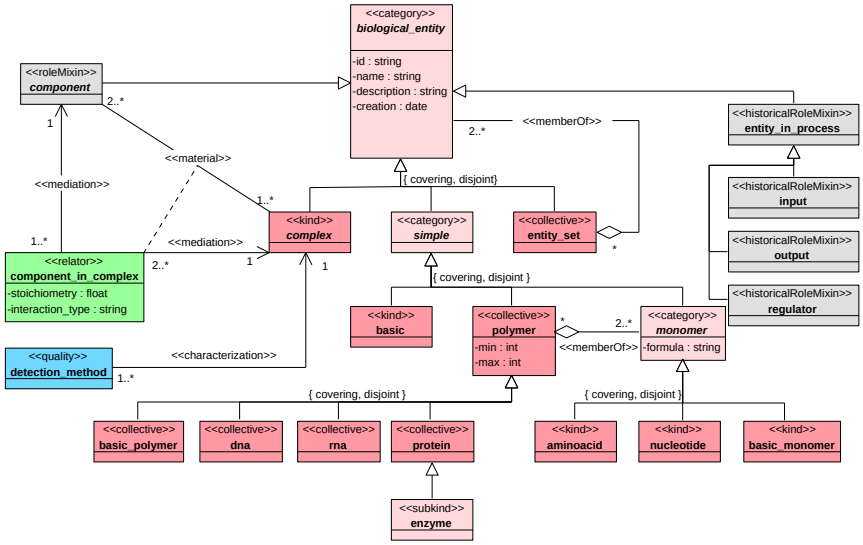
Characterization of biological entities

We stereotyped the `polymer` class as a collective when characterizing the different classes used to identify biological entities. Collectives are constructs made of parts whose role is the same with respect to the whole. Although these parts are modeled in the UML version, it is not clear how the whole and its parts are connected (see Figure 6a). The model allowed us to represent the same entity in multiple ways (e.g., a protein could be represented through both the `Protein_e` class and the `Polymer`

class). In the unpacked version, we thus created the class `monomer` and connected it to the `polymer` class. Then, we reorganized the existing subtypes of single entities by determining whether they are polymers or monomers. This change facilitates the identification of the parts (monomer) that compose the collective (polymer). This change also removes the possibility of representing the same entity in more than one way (e.g., a protein is now represented through the `protein` class, which is a subtype of polymer).



(a) UML excerpt



(b) OntoUML excerpt

Figure 6: Model excerpts for characterizing biological entities.

The "flat" semantics of UML does not take into account the identity and rigidity dimensions. UML represents objects with identity (e.g., «kind») in the same way as those without identify (e.g., «category» or «roleMixin»). Similarly, for objects whose instances are rigid (e.g., «kind») or anti-rigid (e.g., «role»). Our analysis shows how these aspects affect conceptual clarity. In the OntoUML version, we can identify the core components and characterize their changes that result from modifications of their internal properties or external interactions (see of Figure 6b).

It is clearer that it is the `Protein_e` class that gives identity to the protein, not the different roles the protein plays in the processes. The OntoUML model clarifies that a biological entity *is not* a regulator, but *acts* as a regulator.

The new characterization of biological entities provides a clearer distinction between them. For instance, the UML model characterized proteins with their own class, called `Protein_e`. However, this model also characterized polymers with a class. The problem is that proteins are polymers, but the classes that represent them are not linked in any way. This has implications at the instance level: should we instantiate a protein as a `Protein_e`, as a `Polymer`, or as both types? While an initial answer might be to use the `Protein_e` class because its only purpose is to model proteins, this approach would hinder the fact that proteins are polymers, violating the conceptual modeling principle of making implicit knowledge explicit. The OntoUML characterization makes the fact that proteins are polymers explicit.

The UML representation required OCL rules to avoid situations where polymers are made of other polymers. Furthermore, what are the exact classes that can form polymers? The answer to this question is in the UML model, but requires implicit knowledge regarding genomics. The OntoUML makes this knowledge explicit by creating the `monomer` class and linking it to the `polymer` class. The new model identifies the types of polymers that need to be described (DNA, RNA, proteins, and basic polymers) and the atomic component, or monomer, that creates them (nucleotides, amino acids, or basic polymers, respectively).

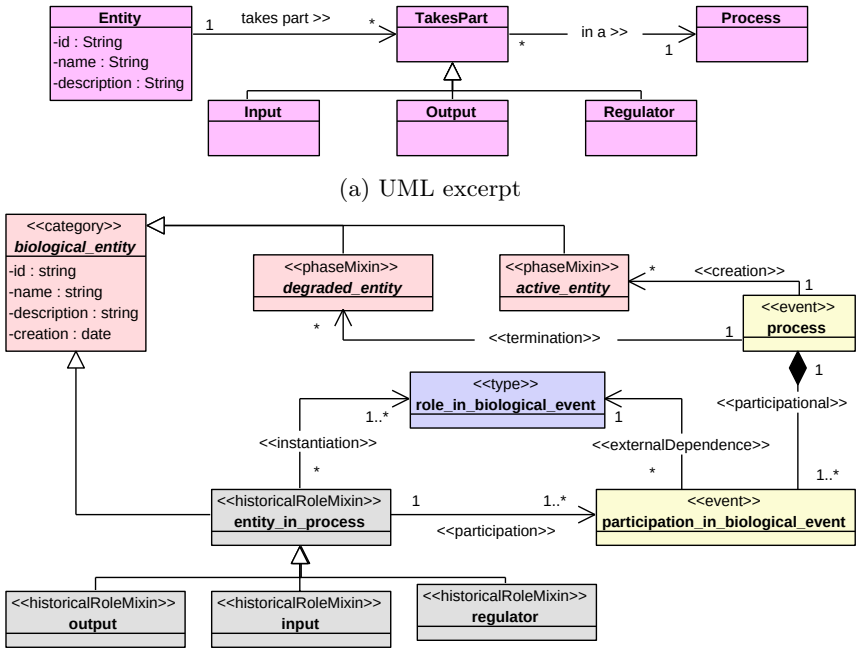
Changes in biological entities over time

The use of the «phase» stereotype in the OntoUML model enriches the representation of the effects caused by events (see Figure 7b). In the UML version, an entity can act as an `Input`, an `Output`, or a `Regulator`. In the OntoUML version, there is an additional dimension that allows us to indicate whether the entity has been degraded. The following examples illustrate what can be modeled using this approach: i) an entity that is degraded as a result of a process; ii) an entity that is created as a result of a process; iii) an entity that is modified as a result of a process; or iv) an entity that is degraded as a result of regulating a process. This change in the state of an entity (i.e., whether an entity is degraded) could not be modeled without the inclusion of the «phase» stereotype. In the OntoUML model, this clarifies that the changes of `biological_entities` in our bodies result from processes. In contrast, it is not clear how to model the degradation of entities with the UML model (see Figure 7a).

The creation of the `active_entity` and `degraded_entity` phases provides additional mechanisms to ensure the correctness of the model. For instance, we can explicitly specify a constraint stating that enzymes are not degraded when they catalyze processes. That is, they cannot instantiate the `degraded_entity` «phaseMixin» in the same process in which they instantiate that catalyst «historicalRoleMixin». This prevents introducing errors when instantiating and populating the model. Such constraints are difficult to identify in the UML model.

Representation of chemical compounds

Thousands of different chemical compounds that take part in the processes that occur in our body continuously. In UML, they are represented with the `Basic_e`



(a) UML excerpt

(b) OntoUML excerpt

Figure 7: Model excerpts concerning the changes in biological entities over time.

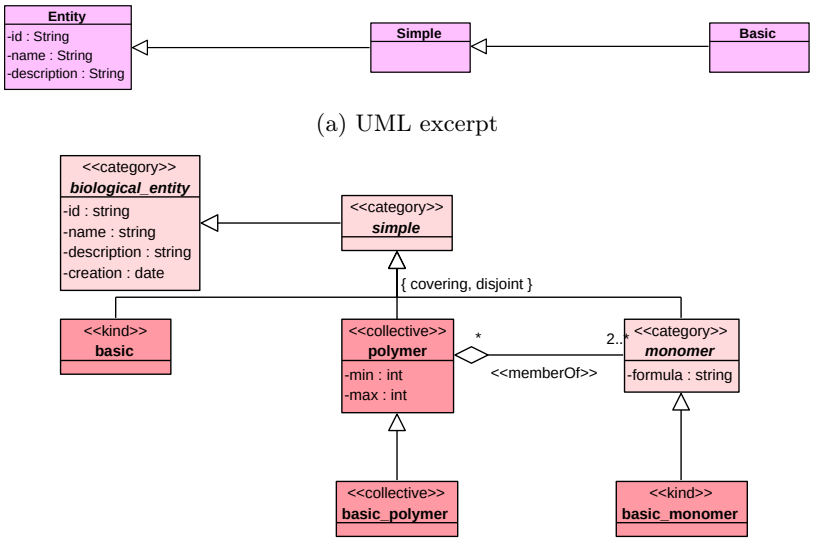
class, which is a type of simple entity (see section (a) of Figure 8). However, this representation is not clear enough to address questions such as: Can a chemical compound be a polymer? What are the monomers of a chemical compound that is a polymer? The stereotypes of OntoUML and the fact that modelers must make such categorization explicit identified the need to model: i) chemical compounds that are not polymers nor monomers; ii) chemical compounds that are polymers; and iii) the monomers of these polymers.

To increase clarity, in OntoUML we created two new classes (see section (b) of Figure 8). The first, `basic_polymer`, is stereotyped with «collective» to represent chemical compounds that are polymers. The second, `basic_monomer`, is stereotyped with «kind» to represent chemical compounds that are monomers. The new representation can differentiate between chemical compounds that are polymers or basic elements; e.g., water is a chemical compound, but not a polymer; maltose is a chemical compound that is a polymer made of the glucose monomer.

Empirical evaluation

Methods

Ontological unpacking is a procedure that costs time and effort. However, we aim to evaluate if its benefits in terms of a better explanation of a complex domain and to justify the process. Our goal is to analyze differences between representing a conceptual model with UML and OntoUML. The purpose is to analyze pros and cons of OntoUML, with respect to usability; the adopted point of view is that of Computer Engineering students that are learning Model-Driven Development (MDD) in their curriculum. We organized our empirical evaluation objectives using



(a) UML excerpt

(b) OntoUML excerpt

Figure 8: Model excerpts concerning the representation of chemical compounds.

the Goal Question Metric template for goal definition following the guidelines for reporting software engineering experiments in [41].

Hypothesis development

ISO 25000 [42] defines usability in terms of effectiveness, efficiency, and satisfaction as "*the degree to which specified users can achieve specified goals with effectiveness in use, efficiency in use and satisfaction in use in a specified context of use*". In this paper, we use this definition of usability to specify our research questions:

- **RQ1**: Is effectiveness in the conceptual modeling interpretation affected by the model notation?
- **RQ2**: Is efficiency in the conceptual modeling interpretation affected by the model notation?
- **RQ3**: Is satisfaction in the conceptual modeling interpretation affected by the model notation?

The research questions lead to define three null hypotheses to be tested throughout the experiment:

- H_{01}: Effectiveness analyzing a conceptual model expressed in OntoUML is the same as with UML.
- H_{02}: Efficiency analyzing a conceptual model expressed in OntoUML is the same as with UML.
- H_{03}: Satisfaction analyzing a conceptual model expressed in OntoUML is the same as with UML.

Factor, response variables and metrics

The *factor* used in the experiment corresponds to the conceptual modeling notation. It presents two levels: the control treatment (i.e., UML notation), and the target

treatment (i.e., OntoUML notation). We choose UML as control treatment because this notation is known by the subjects before the experiment.

We then defined one *response variable* for each null hypothesis to be tested. The first response variable is *Effectiveness*, defined by the IEEE dictionary [43] as "*the accuracy and completeness with which users achieve specified goals*". Effectiveness is measured through a questionnaire (*model questionnaire*) whose questions investigate the meaning of the elements represented in several parts of a conceptual model. The model questionnaire can be appreciated in [44]. Each answer has two possible values: correct (1) or failure (0). Questions are divided into three groups: questions related to entities, questions related to events, and questions related to entities involved in events. For each of such groups we defined a metric, calculated as the sum of values associated to its answers. For example, if the metric events is composed of three questions of which only two have been answered correctly, the value for events is 2. We also analyzed the value of the questions individually, in order to understand what specific elements of the model may affect the effectiveness in different degrees.

The second response variable is *Efficiency*, defined in the IEEE dictionary [43] as "*the degree to which a system or component performs its designated functions with minimum consumption of resources*". We propose measuring the resources in terms of the analyst's time spent in understanding the conceptual model with the purpose to answer the model questionnaire. We have one metric for each group of questions of the model questionnaire (entities, events, and entities in events). The time for each group of questions is calculated as the sum of the time spent to answer each question of that group. For example, if the entities group has three questions and the subject spent respectively 30, 20, and 15 seconds to answer such questions, then the metric for entities is 65. Also here, we additionally analyzed the time to answer each question individually – to identify the questions that required more time.

The third response variable is *Satisfaction*, defined in the IEEE dictionary [43] as "*freedom from discomfort, and positive attitudes towards the use of the product*". As proposed by Davis [45], we measured it using three metrics: Perceived Ease Of Use (PEOU), Perceived Usefulness (PU), and Intention To Use (ITU). The three metrics were measured using a 5-point Likert scale questionnaire named Method Adoption Model (MAM). Based on Moody [46], in the MAM questionnaire, we defined six questions to measure Perceived Ease of Use, eight questions for Perceived Usefulness, and two questions for Intention to Use. The metric for PEOU, PU and ITU is calculated as the addition of the answers for each one of them. Therefore, possible values for PEOU are between 6 and 30, for PU are between 8 and 40, and for ITU are between 2 and 10. We defined a questionnaire for each treatment (UML and OntoUML); questions used the same template in both questionnaires, adapted to each treatment (see questionnaire in [44]).

Subjects

The experiment was carried out with twenty subjects. We asked them to complete a demographic survey to understand their background and mitigate possible validity threats. All of the subjects are computer engineering students in their third year and have a Grade Point Average (GPA) of 7.5. More than 50% of subjects (12 out

of 20) have no previous working experience, and only 25% indicated that they have more than one year of working experience (mostly as junior developers).

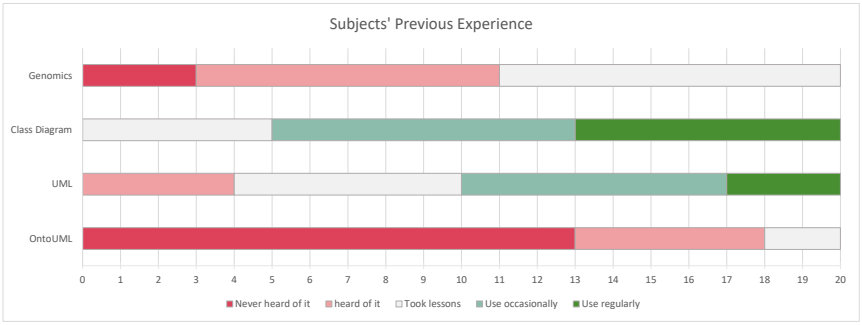


Figure 9: Subjects' previous experience regarding genomics, class diagrams, UML language, and OntoUML language.

See Figure 9 for a visual representation of subjects' experience in the involved topics. All subjects knew about class diagrams and the UML language. The majority took classes on both (only four subjects did not take any class of UML). However, only half of the subjects took classes on genomics, whereas three of them had never heard of it. Likewise, 65% of the subjects had never heard of OntoUML before the experiment, and only two of them had studied it in their classes.

Experiment problems

We are interested in measuring how better a OntoUML schema can explain relevant concepts of a specific domain when compared to its corresponding UML schema. To test this, we gathered a series of questions that are considered important by experts in the domain of metabolic pathways. From a pool of questions proposed by them, we selected 18 questions, to be divided in three groups related to entities (6 questions), events (6) and interaction of entities within events (6). These questions were then distributed into two different problems (P1 and P2), attempting to offer an homogeneous level of difficulty and variety of topics.

Experiment design

The experiment is a within-subjects design (repeated measures) where two factors are applied to all subjects. As a block variable[3] we consider the assigned problem, since we are not interested in analyzing differences between problems, but in analyzing if the type of problem may affect the results. In order to avoid that the order in the used treatments application and problems affects the results, we organized the subjects into four groups. Each group represents a possible combination of problem and treatment. Groups are balanced and subjects are randomly assigned to one group.

[3]A block variable is a variable we are not interested in study but we aim to ensure that is not affecting the results.

Problem	Group	ID	Competency Questions
P1	Entities	1	Polymers are composed of other polymers.
		2	The internal structure of any polymers is homogeneous.
		3	The internal structure of basic biological entities and polymers is the same.
	Events	4	Processes are limited in time.
		5	Pathways must be composed of other pathways.
		6	A process can be decomposed into other events.
	Interaction	7	Every biological entity must participate in at least one process.
		8	Biological entities can take part in pathways.
		9	A protein can take the roles of input, output, and regulator in the same process.
P2	Entities	10	Some polymers are composed of nucleotides.
		11	Every enzyme is a polymer.
		12	Some basic biological entities can be polymers also.
	Events	13	Every event must have a preceding event.
		14	Pathways can be composed of other pathways.
		15	Events occur in a specific time interval.
	Interaction	16	Biological entities can be created and destroyed as a result of a process.
		17	Biological entities can participate in multiple processes.
		18	A protein can take the role of input in different processes.

Table 1: Questions posed to subjects, clustered by Problem number and group (regarding entities, events, or their interaction)

Experiment procedure

After collecting demographic surveys from subjects, we run two teaching sessions respectively on the theory and practice of UML and OntoUML. Each class lasted 45 minutes. After each class we asked the subjects to complete a knowledge assessment questionnaire to prove their understanding of the received information. The test was composed of eight questions regarding a model (respectively drawn with UML or OntoUML) that concerned a topic not related to genomics. Once we ascertained that the knowledge of all participants was sufficient to be included in the study, we distributed to them the questionnaires with questions on the models (i.e., Problems P1 and P2). Subjects used alternatively one of the two treatments for answering questions; specifically, participants used the UML model and its corresponding OntoUML model available in [44]. These are fragments of the models shown in Figures 3 and 5; the represented domain segment was carefully selected so that it is representative of the full model. Then, they also filled in one MAM questionnaire for each used formalism. The detailed workflows of each of the four groups is provided in Table 2.

Group n°	First task	Second Task	Third Task	Fourth Task
1	Problem P1 (UML)	PEOU-PU-ITU (UML)	Problem P2 (OntoUML)	PEOU-PU-ITU (OntoUML)
2	Problem P2 (UML)	PEOU-PU-ITU (UML)	Problem P1 (OntoUML)	PEOU-PU-ITU (OntoUML)
3	Problem P1 (OntoUML)	PEOU-PU-ITU (OntoUML)	Problem P2 (UML)	PEOU-PU-ITU (UML)
4	Problem P2 (OntoUML)	PEOU-PU-ITU (OntoUML)	Problem P1 (UML)	PEOU-PU-ITU (UML)

Table 2: Groups organization.

The data analysis performed on the experiment outcomes was based on descriptive data and statistical analysis. We chose to report descriptive data using box-and-whisker plots to illustrate the differences regarding the treatments of the design

variable. Descriptive data helps graphically identifying possible differences between treatments or among levels. As a statistical test to identify significant differences between treatments and among replications we used a mixed model. The assumption for applying the mixed model is normality of residuals, which can be tested with the Shapiro-Wilk test applied to the residuals automatically calculated during the application of the mixed model test [47]. When the p-value is less than 0.05, we can reject the null hypothesis, which means that there are significant differences for the variable. We used Cohen's d [48] to calculate the effect size in those variables with significant differences (variables whose p-value with the mixed model is less than 0.05). Cohen's d is defined as the difference between two means divided by a standard deviation of the data. According to [48], the meaning of the effect size is as follows: more than 0.8 is a large effect; from 0.79 to 0.5 is a moderate effect; from 0.49 to 0.2 is a small effect. Using the mixed model, we cannot calculate power statistically (independently of the statistical tool used in the analysis). However, we used G*Power [49], finding that, for a repeated measurement statistical test, we need a sample size of 16 units for an effect size of 0.8 (large effect) to get a power of 80%. Since we have 20 sample units, we can state that we have enough power to conduct the statistical analysis.

Results

This section describes the results for each response variable. The *Effectiveness* variable has been measured for the entities, for the events, and for the participation of entities in events. Figure 10A shows the box plot for the effectiveness of the entities. These descriptive results show that OntoUML yields better effectiveness than UML; median, first quartile and third quartile are clearly better for OntoUML. The line that connects both treatments represents the averages. Figure 10B shows the box plot for the effectiveness of the events. This is very similar to the effectiveness of the entities, median, first and third quartiles are better for OntoUML. Figure 10C shows the box plot for the effectiveness of the participation of entities in events. In this case, all median, first and third quartile are the same for both treatments.

Table 3 shows the statistical analysis of effectiveness for their different metrics. We detail the results for entities, events and entities in events. Entities and events yield significant results as p-values are lower than .05. The size of the effect is large, which means that these differences are important. We have not identified significant differences in the Method*Problem for these metrics (see Interaction column in Table 3), which means that the problems used in the experiment are not affecting the results. To conclude, we can reject H_{01} for the Entities and Events metrics, which means that effectiveness then analyzing a conceptual model with OntoUML is better than with UML for those metrics.

Next, we analyze the results for the variable *Efficiency* considering its three metrics, namely, entities, events and entities in events. Figure 10D shows the box plot for the efficiency of the entities. Medians, first and third quartile are higher for OntoUML. This pattern is repeated also for the other two metrics efficiency events and efficiency entities in events, as shown in Figures 10E and 10F, respectively.

Table 4 shows the statistical results after applying the Mixed Model to the data. Importantly, all metrics yield significant results (p-values $<$ .05). This appears in

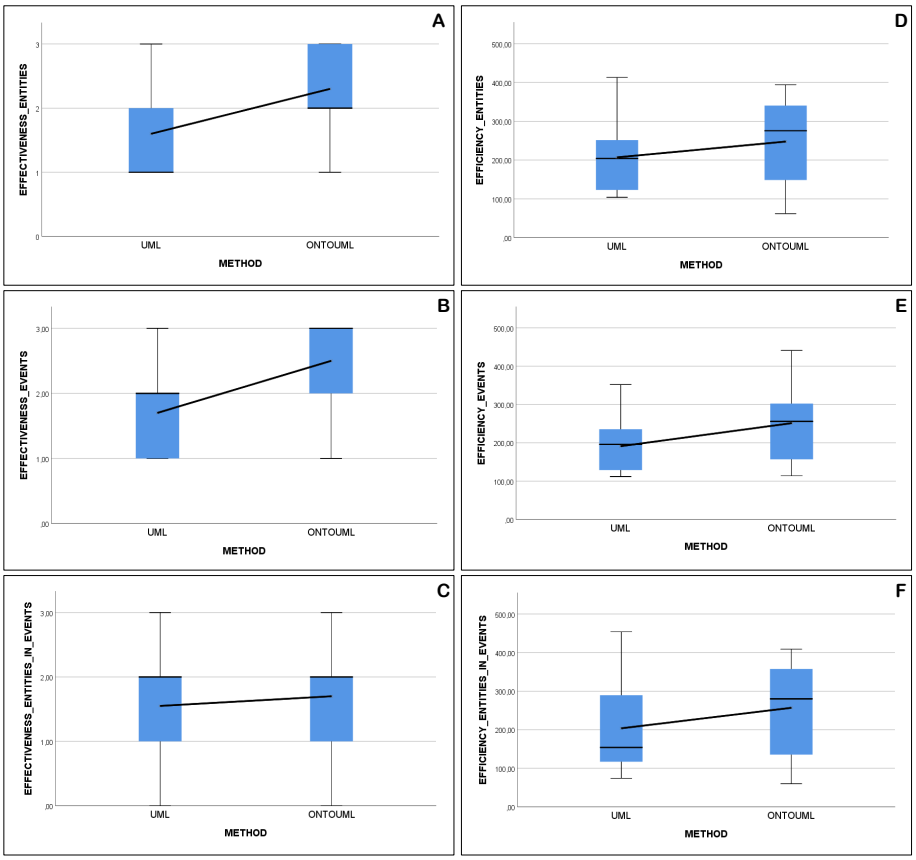


Figure 10: Box and whiskers plots for the effectiveness of entities (Panel A); the effectiveness of events (Panel B); the effectiveness of the interaction between entities and events (Panel C); the efficiency of entities (Panel D); the efficiency of events (Panel E); the efficiency of the interaction between entities and events (Panel F).

Table 3: Data analysis results for effectiveness metrics.

	Treatment	Interaction	Mean	Effect Size
ENTITIES	**.001	.112	UML: 1.6 OntoUML: 2.3	**.98**
EVENTS	**.001	.388	UML: 1.7 OntoUML: 2.5	**1.2**
ENTITIES IN EVENTS	0.587	.285	UML: 1.55 OntoUML: 1.7	-

the general metrics of entities, events and entities in events. Thus, OntoUML requires significantly more time when compared to UML. Regarding Method*Problem interaction, we obtained no significant results (see Interaction column in Table 4), so we can state that the problem is not affecting the results.

Thus, we can reject H_{02} for all metrics, which means that the efficiency, when analyzing a conceptual model with UML, is better than with OntoUML.

Table 4: Data analysis results for efficiency metrics.

	Treatment	Interaction	Mean	Effect Size
ENTITIES	**.006	.165	UML: 206.95 OntoUML: 247.65	.4
EVENTS	**.000	.731	UML: 191.25 OntoUML: 251.4	.71
ENTITIES IN EVENTS	**.001	.468	UML: 203.65 OntoUML: 256.85	.44

Last, we analyze the results for the variable *Satisfaction* considering its three metrics: perceived ease of use, perceived usefulness and intention to use. Figure 11A shows the box plot for perceived usefulness. Median, first and third quartile show a higher satisfaction for UML. This pattern is also repeated for perceived usefulness (Figure 11B and intention to use (Figure 11C).

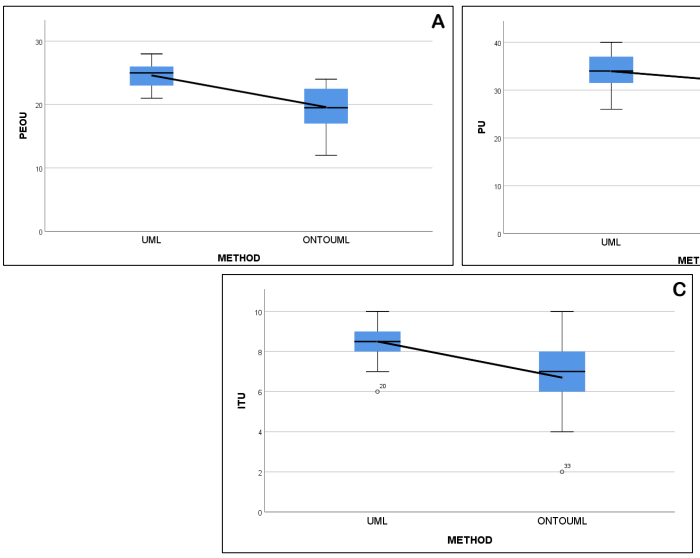


Figure 11: Box and whiskers plot for perceived ease of use (Panel A); perceived usefulness (Panel B); and intention to use (Panel C).

Table 5 show the results of the Mixed Model for the metrics of satisfaction. All three metrics show significant results (p-value $<.05$), yielding UML a better average rather than OntoUML. This means that analysts working with UML yield significantly better satisfaction than analysts working with OntoUML.

Table 5: Data analysis results for satisfaction metrics.

	Treatment	Interaction	Mean	Effect Size
PEOU	**.005	.843	UML: 8.5 OntoUML: 6.7	**1.1**
PU	**.003	.923	UML: 33.95 OntoUML: 30.9	.78
ITU	**.005	.843	UML: 8.5 OntoUML: 6.7	**1.1**

Thus, we can state that we can reject H_{03} for all metrics, which indicates that satisfaction analyzing a conceptual model with UML is better than with OntoUML.

Discussion

With regard to *Effectiveness* (H_{01}), the empirical analysis allowed us to conclude that OntoUML was more effective in conveying the genomics domain to the study participants, backed by a relevant statistical significance for the Entities and Events groups. Specifically, we found that: i) Entity-related questions were answered more successfully with OntoUML (likely because UFO contains stereotypes that helped clarifying important principles, such as rigidity); ii) Events-related questions were also answered more successfully with OntoUML, with a more relevant difference (the ontological foundation of events presented in UFO may have helped participants to capture relevant details regarding event-related information); iii) Questions related to the Interaction between events and entities were answered more successfully with OntoUML by a very small fraction. A number of interesting aspects can be discussed:

- Conceptual modeling aims to make implicit concepts explicit. From a biological point of view, events are clearly limited in time. However, in the UML model (Figure 3), the temporal limitations of a process are left implicit. Based on our ontological analysis, such information was extracted and explicitly represented by means of the «event» stereotype. This particular difference was observed in questions Q4 (OntoUML: 90%, UML: 30%) and Q15 (OntoUML: 80%, UML: 10%), where OntoUML allowed to respond correctly to a higher percentage of participants.
- A simple explanation of the participation of entities in the processes is provided by the UML model (Figure 3), whereas the OntoUML version (Figure 5) provides a more complex and detailed explanation, as OntoUML allows us to analyze the *mereology of events*. In particular, Q6 was answered with a higher score using OntoUML (70%) instead of UML (40%), likely because the UML model left the individual participation of chemical compounds in reactions implicit. Unexpectedly, Q17, also concerning events mereology (specifically, the participation in multiple processes), was instead better answered by means of the UML model (80%) rather than with OntoUML (%20). In this case, respondents were probably confused by the complexity of the representation, which should be object of further study and evaluation.
- OntoUML expresses the «phase» stereotype, exploiting the principle of rigidity [50], which clarifies the fact that chemical compounds and biological-related substances are created and destroyed as a result of chemical reactions. Q16 regards this aspect and showed a significant difference between the two formalisms (OntoUML 90% vs. UML 30%), demonstrating the higher capability of OntoUML of explaining such principle.

The *Efficiency* assessment was measured through H_{02}; the null hypothesis could be rejected for all groups, as OntoUML required longer response times. This was likely due to the complexity of OntoUML and the very limited experience of participants with it. Our initial expectation suggested that a complex domain explained

through a more complete and explicit model would also translate into shorter answering times. Instead, the evaluation showed that OntoUML required more time to participants to be able to answer questions based on it.

The user *Satisfaction* assessment, tested through H_{03}, showed that – for all groups – OntoUML was, in general, less appreciated by users, as perceived as more complex. This language was completely new to the participants, who lacked any experience using it. They were hesitant to learn and use a novel modeling language, especially a complex one, in a short amount of time. However, the results indicate that performances, in terms of effectiveness, were better using OntoUML. As shown in other works [15, 16], subjects need generally more time to properly understand the paradigm of ontological conceptual modeling and, specifically, the OntoUML language intricacies. Therefore, further experimentation on PEOU, PU, and ITU should be repeated with subjects that have received a longer training.

To summarize, the practical adoption of the ontological unpacking method is currently hindered by the long learning curve of the formalism on the part of users. It is apparent that a previous background in OntoUML greatly facilitates the use of the models. Thus more effort should be dedicated to the teaching and use of this formalism. At the same time, the design of an OntoUML model typically takes longer than a simpler UML model. Nevertheless, the shared objective of a better interdisciplinary exchange that is enabled by this method should justify the overhead in terms of efforts. On one hand, domain experts should be interested in providing more complete and unambiguous models. On the other hand, users should be interested in artifacts that convey information more clearly and correctly.

Validity

We considered four types of threats (i.e., to conclusion validity, to internal validity, to construct validity, and to external validity), as defined for quasi-experimentation by Campbell and Stanley [51] and extended by Cook and Campbell [52].

Threats to *conclusion validity* [41] affect the ability to obtain correct conclusions about relations between the treatment and the experiment outcome. Typical threats include: i) the *low statistical power*, which here was mitigated by using G*Power [49] to estimate the minimum sample size needed for achieving statistical significance; ii) the *reliability of measures*, which was mitigated by asking domain experts to double-check the list of questions for proper wording; iii) the *random irrelevancies in experimental setting*, which was mitigated by making sure that all participants were comfortable in the classroom, were never interrupted, and did not collaborate with each other; iv) the *random heterogeneity of subjects*, which was mitigated by choosing a set of participants from the same curriculum, with an homogenous knowledge level on Class Diagrams and without previous knowledge on OntoUML and genomics (see Figure 9). To level out possible differences among participants' preparation, two classes of the same duration were given on both UML and OntoUML.

Threats to *internal validity* [41] affect the experimental factor (i.e., the modeling formalism) with respect to causality. Several of such threats can be mitigated by performing a multiple-groups experiment (vs. single group). We thus carried out our experiment with four groups; to deal with *interactions with selection* we carefully

designed the experiment such that each group applied each treatment (UML and OntoUML) to two similar problems (P1 and P2) in different order (see Table 2). No interactions between the groups was allowed.

Threats to *construct validity* [41] can create results that are not generalizable in the form of a theory behind the experiment. First, we considered the design-related threats. To mitigate a possible *inadequate pre-operational explication of constructs*, we gave two classes about the involved treatments of the same duration – adequately introducing UML and OntoUML. Threats of *interaction of different treatments* were mitigated by the four-groups setup, which was also useful to deal with *interactions with selection* (internal validity). *Restricted generalizability across constructs* was addressed by measuring Effectiveness and Efficiency. Conclusions were drawn taking both into consideration. Then, we considered social threats. Participants, possibly developing *evaluation apprehension*, were reassured that no marks would be derived from the experiment. To reduce any *experimenter expectancies* that could bias the results, the raised questions were prepared by external domain experts.

Finally, threats to *external validity* [41], especially when conducting the experiment on students, can limit the overall generalizability of results outside of the specific context. However, using students as participants is known to be a valid simplification of reality needed in laboratory contexts [53].

In the future we plan to run more general evaluations, using other models (possibly in other life sciences domains other than genomics) and involving a larger number of participants, thereby allowing us to consider also other aspects that were here ignored (e.g., demography, learning styles, previous general modeling experience).

Conclusion

The modeling of the human genome is an effort to understand life itself through the development of a conceptual model. This research has implications for both researchers and scientists. First, recognizing the complexity of this domain shows the importance of representing the human genome by a model that supports a shared understanding. Second, by making the ontological clarity of the conceptual model explicit, it is possible for the model to have a solid foundation. For example, for events, we characterized how they can be decomposed into more specific events, how they can be identified by the participation of biological entities in processes (i.e., a specific type of event), and how they relate to each other. Moreover, having this model represented in OntoUML allows us to benefit from the existing support for this language in terms of formal verification, validation, and reasoning by automatically generating an OWL specification for the model. All such advantages motivated our work which first analyzed an existing model of pathways designed with traditional modeling techniques, and then proposed an enriched version that resolves unclear and ambiguous areas of the domain.

Further work will add the OntoUML notions of situation and disposition. The situation represents transformations from a portion of reality to another one through events. Dispositions capture properties intrinsically dependent on objects that can be manifested under specific situations. These concepts are important in genomics and precision medicine because they enable the representation of diseases and pathways using situations and altered functions of modified proteins as dispositions. We

are confident that such additions will draw human genome researchers closer to adopting the proposed models. Moreover, further ontological unpacking will be applied to the other conceptual views of the CSHG, regarding structural, variations, transcription, and proteome aspects.

This work aims to reinforce conceptual models as a practical way for domain experts and computer scientists to share the knowledge needed to develop genomic information systems and support processing heterogeneous genomics data.

Funding
This work has been developed with the financial support of the Generalitat Valenciana and the Valencian Innovation Agency under the projects MICIN/AEI/10.13039/501100011033, CIPROM/2021/023, ACIF/2021/117 and INNEST/2021/57 and co-financed with ERDF.

Abbreviations
BFO: Basic Foundational Ontology
CSHG: Conceptual Model of the Human Genome
DOLCE: Descriptive Ontology for Linguistic and Cognitive Engineering
GPA: Grade Point Average
ITU: Intention To Use
MAM: Method Adoption Model
MDD: Model-Driven Development
OCL: Object Constraint Language
OntoUML: an ontologically well-founded language for ontology-driven CM; it is built as a UML extension based on UFO
PEOU: Perceived Ease Of Use
PU: Perceived Usefulness
UFO: Unified Foundational Ontology
UML: Unified Modeling Language

Availability of data and materials
All the materials used for introducing the topics to the subjects and to assess their understanding are available on Zenodo at https://doi.org/10.5281/zenodo.6616114.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Acknowledgements
Not applicable.

Authors' contributions
Ontological unpacking: AGS and OP built the initial UML model; AGS and GG built the OntoUML model and performed the ontological unpacking analysis.
Empirical evaluation: AGS and AB designed and carried out the experiment; AGS prepared the data; and IP performed the statistical analysis.
Management: GG, OP, and VS supervised the research. AB and AGS performed the research and wrote the first draft of the manuscript. All authors participated in the discussions and finalized the manuscript.

Author details
[1]PROS Research Center & VRAIN Research Institute, Universidad Politècnica de València, Camino de Vera S/N, 46021 Valencia, Spain. [2]Department of Electronics, Information and Bioengineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133, Milan, Italy. [3]Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Universitá, 1, 39100, Bolzano, Italy. [4]Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Drienerlolaan 5, 7522 Twente, Netherlands. [5]J. Mack Robinson College of Business, Georgia State University, 33 Gilmer St SE, 30303 Atlanta, Georgia, United States.

References
1. Smith, B., Williams, J., Steffen, S.-K.: The ontology of the gene ontology. In: AMIA Annual Symposium Proceedings, vol. 2003, p. 609 (2003). American Medical Informatics Association
2. Gaudet, P., Dessimoz, C.: Gene ontology: Pitfalls, biases, and remedies. In: Dessimoz, C., Škunca, N. (eds.) The Gene Ontology Handbook. Methods in Molecular Biology, pp. 189–205. Springer
3. Schulz, S., *et al.*: Strengths and limitations of formal ontologies in the biomedical domain. Revista electronica de comunicacao, informacao & inovacao em saude : RECIIS **3**(1), 31–45 (2009)
4. Olivé, A.: Conceptual Modeling of Information Systems. Springer, Berlin Heidelberg (2007)

5. Pastor, O., Gómez, J., Insfrán, E., Pelechano, V.: The OO-method approach for information systems modeling: from object-oriented conceptual modeling to automated programming. Information Systems **26**(7), 507–534 (2001)
6. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, Berlin Heidelberg (2007)
7. Pastor, O.: Conceptual modeling of life: beyond the homo sapiens. In: International Conference on Conceptual Modeling, pp. 18–31 (2016). Springer
8. Pastor, O., *et al.*: Using conceptual modeling to improve genome data management. Briefings in Bioinformatics **22**(1), 45–54 (2021)
9. García S., A., *et al.*: Towards the Understanding of the Human Genome: A Holistic Conceptual Modeling Approach. IEEE Access **8**, 197111–197123 (2020). Conference Name: IEEE Access
10. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. PhD thesis, University of Twente (January 2005)
11. Guizzardi, G., *et al.*: Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. Applied Ontology **10**(3-4), 259–271 (2015). Publisher: IOS Press
12. Guizzardi, G., Bernasconi, A., Pastor, O., Storey, V.C.: Ontological unpacking as explanation: The case of the viral conceptual model. In: International Conference on Conceptual Modeling, pp. 356–366 (2021). Springer
13. Bernasconi, A., Canakoglu, A., Pinoli, P., Ceri, S.: Empowering virus sequence research through conceptual modeling. In: International Conference on Conceptual Modeling, pp. 388–402 (2020). Springer
14. Garcia, A., Guizzardi, G., Pastor, O., Storey, V.C., Bernasconi, A.: An ontological characterization of a conceptual model of the human genome. In: International Conference on Advanced Information Systems Engineering (CAISE) Forum, pp. 27–35 (2022). Springer
15. Verdonck, M., Gailly, F., Pergl, R., Guizzardi, G., Martins, B., Pastor, O.: Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study. Information Systems **81**, 92–103 (2019)
16. Verdonck, M., Gailly, F., de Cesare, S.: Comprehending 3d and 4d ontology-driven conceptual models: An empirical study. Information Systems **93**, 101568 (2020)
17. Kalibatiene, D., Miliauskaitė, J.: A systematic mapping with bibliometric analysis on information systems using ontology and fuzzy logic. Applied Sciences **11**(7), 3003 (2021)
18. Keet, C.M., Khan, Z.: Foundational ontologies: From theory to practice and back. Journal of Knowledge Structures and Systems **3**(1) (2022)
19. Chen, P.P.-S.: The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS) **1**(1), 9–36 (1976)
20. Mylopoulos, J.: Conceptual modelling and telos. conceptual modelling, databases, and case: An integrated view of information system development. New York: John Wiley & Sons **49**, 68 (1992)
21. Guizzardi, G., Wagner, G., Almeida, J.P.A., Guizzardi, R.S.: Towards ontological foundations for conceptual modeling: The unified foundational ontology (ufo) story. Applied ontology **10**(3-4), 259–271 (2015)
22. Arp, R., Smith, B., Spear, A.D.: Building Ontologies with Basic Formal Ontology. Mit Press, Cambridge (2015)
23. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: The WonderWeb Library of Foundational Ontologies and the DOLCE ontology. WonderWeb Deliverable D18, final report (vr. 1.0, 31-12-2003) (2003)
24. Partridge, C.: Business objects. Re-enginnering for re-use (2nd ed.). UK: The BORO Centre (2005)
25. Booch, G., Jacobson, I., Rumbaugh, J., *et al.*: The unified modeling language. Unix Review **14**(13), 5 (1996)
26. Flowers, R., Edeki, C.: Business process modeling notation. International Journal of Computer Science and Mobile Computing **2**(3), 35–40 (2013)
27. Waldemarin, R.C., de Farias, C.R.: Obo to uml: Support for the development of conceptual models in the biomedical domain. Journal of Biomedical Informatics **80**, 14–25 (2018)
28. Pastor, O., *et al.*: Model-based engineering applied to the interpretation of the human genome. In: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol. 6520, pp. 306–330. Springer, ??? (2011)
29. Reyes Román, J.F., *et al.*: Applying conceptual modeling to better understand the human genome. In: International Conference on Conceptual Modeling (ER), vol. 9974 LNCS, pp. 404–412 (2016). Springer
30. Román, R., José Fabián: Design and development of a genomic information system based on a holistic conceptual model of the human genome. Ph.d. thesis, Polythecnic Unversity of Valencia (March 2018). doi:10.4995/Thesis/10251/99565. Accepted: 2018-03-22
31. Palacio, A.L., *et al.*: Genomic Information Systems applied to Precision Medicine: Genomic Data Management for Alzheimer's Disease Treatment. In: International Conference on Information Systems Development (ISD) (2018). https://aisel.aisnet.org/isd2014/proceedings2018/eHealth/6
32. Palacio, A.L., *et al.*: Towards an effective medicine of precision by using conceptual modelling of the genome. In: Proceedings - International Conference on Software Engineering, pp. 14–17. IEEE Computer Society, New York, New York, USA (2018)
33. Román, J.F.R., *et al.*: Use of GeIS for early diagnosis of alcohol sensitivity. In: BIOINFORMATICS 2016 - 7th International Conference on Bioinformatics Models, Methods and Algorithms, Proceedings; Part of 9th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2016, pp. 284–289. SCITEPRESS - Science and and Technology Publications, ??? (2016)
34. León Palacio, A., *et al.*: Genomic Data Management in Big Data Environments: The Colorectal Cancer Case. In: Woo, C., *et al.*(eds.) Advances in Conceptual Modeling vol. 11158, pp. 319–329. Springer, Cham (2018)
35. Navarrete-Hidalgo, M., *et al.*: Design and Implementation of a Geis for the Genomic Diagnosis using the

SILE Methodology. Case Study: Congenital Cataract:. In: Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 267–274. SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal (2018)
36. Reyes Román, J.F., *et al.*: GenesLove.Me 2.0: Improving the Prioritization of Genetic Variations. In: Damiani, E., *et al.*(eds.) Evaluation of Novel Approaches to Software Engineering vol. 1023, pp. 314–333. Springer, Cham (2019)
37. Iñiguez-Jarrín, C., *et al.*: GenDomus: Interactive and collaboration mechanisms for diagnosing genetic diseases. In: ENASE 2017 - Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 91–102 (2017)
38. Almeida, J.P.A., *et al.*: Events as Entities in Ontology-Driven Conceptual Modeling. In: Laender, A.H.F., *et al.*(eds.) Conceptual Modeling. Lecture Notes in Computer Science, pp. 469–483. Springer, Cham (2019)
39. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM **26**(11), 832–843 (1983)
40. Guizzardi, G., Wagner, G., de Almeida Falbo, R., Guizzardi, R.S., Almeida, J.P.A.: Towards ontological foundations for the conceptual modeling of events. In: International Conference on Conceptual Modeling, pp. 327–341 (2013). Springer
41. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Springer, Berlin Heidelberg (2012)
42. ISO/IEC: Iso/iec 25000 - software engineering - software product quality requirements and evaluation (square) - guide to square (2010)
43. IEEE: IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries., Institute of Electrical and Electronics Engineers. New York, EE.UU. (1991)
44. García S., A., Bernasconi, A.: UML Vs OntoUML Analysis Results [Data Set]. doi:10.5281/zenodo.6616114. Last accessed July 20th, 2022
45. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q. **13**(3), 319–340 (1989)
46. Moody, D.L.: The method evaluation model: a theoretical model for validating information systems design methods (2003)
47. Meyers, L.S., Gamst, G., Guarino, A.J.: Applied Multivariate Research : Design and Interpretation. SAGE Publications, Thousand Oaks (2006). http://www.loc.gov/catdir/toc/ecip0510/2005009519.html
48. Cohen, L.: Statistical Power Analysis for the Behavioral Sciences, 2nd. edition edn. Lawrence Earlbaum Associates, New York, New York (1988)
49. Faul, F., Erdfelder, E., Lang, A.-G., Buchner, A.: G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behavior Research Methods **39**(2), 175–191 (2007)
50. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. CTIT, Centre for Telematics and Information Technology, Twente, Netherlands (2005)
51. Campbell, D.T.: Experimental and quasi-experimental designs for research on teaching. Handbook of research on teaching **5**, 171–246 (1963)
52. Cook, T.D., Campbell, D.T., Day, A.: Quasi-experimentation: Design & Analysis Issues for Field Settings vol. 351. Houghton Mifflin Boston, ??? (1979)
53. Falessi, D., Juristo, N., Wohlin, C., Turhan, B., Münch, J., Jedlitschka, A., Oivo, M.: Empirical software engineering experts on the use of students and professionals in experiments. Empirical Software Engineering **23**(1), 452–489 (2018)