Available online at www.sciencedirect.com

ScienceDirect

Information Systems 00 (2023) 1–23

Journal Logo

Assessing the value of ontologically unpacking a conceptual model for human genomics

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

Abstract

Although the knowledge about human genomics is available to all scientists, information about this scientific breakthrough can often be difficult to fully comprehend and share. A Conceptual Schema of the Human Genome was previously developed to assist in describing human genome-related knowledge, by representing a holistic view of the relevant concepts regarding its biology and underlying mechanisms. This model should become helpful for any researcher who works with human genomics data. We, therefore, perform the process of *ontological unpacking* on a portion of the model, to facilitate domain understanding and data exchange among heterogeneous systems. The ontological unpacking is a transformation of an input conceptual model into an enriched model based on a foundational ontology. The preliminary analysis and enrichment process are supported by the ontological conceptual modeling language OntoUML, which has been applied previously to complex models to gain ontological clarity. The value of the used method is first assessed from a theoretical point of view: the transformation results in significant, diverse modeling implications regarding the characterization of biological entities, the representation of their changes over time, and, more specifically, the description of chemical compounds. Since the ontological unpacking process is costly, an empirical evaluation is conducted to study the practical implications of applying it in a real learning setting. A particularly complex domain such as metabolic pathways is either described by adopting a traditional conceptual model or explained through an ontologically unpacked model obtained from a traditional model. Our research is evidence that including a strong ontological foundation in traditional conceptual models is useful. It contributes to designing models that convey biological domains better than the original models.

Keywords: Ontological Unpacking, Conceptual Modeling, Foundational Ontology, OntoUML, Genomics, Metabolic Pathways, Data Explanation

*AGS and AB should be considered equal contributors.

Email addresses: algarsi3@pros.upv.es (Alberto García S.), abernas@upvnet.upv.es/anna.bernasconi@polimi.it (Anna Bernasconi), g.guizzardi@utwente.nl (Giancarlo Guizzardi), opastor@dsic.upv.es (Oscar Pastor), vstorey@gsu.edu (Veda C. Storey), jopana@upv.edu.es (Ignacio Panach)

[1]PROS Research Center & VRAIN Research Institute, Polytechnic University of Valencia, Spain

[2]Department of Electronics, Information and Bioengineering, Politecnico di Milano, italy

[3]Semantics, Cybersecurity & Services Group (SCS), Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands

[4]J. Mack Robinson College of Business, Georgia State University, United States

1. Introduction

Modeling human genomics knowledge is a fascinating and extremely important area of research due to its potential to impact all of mankind through improved treatments and possibly, the removal of diseases. In essence, this modeling contributes to understanding life itself. Unfortunately, progressing research on human genomics is generally challenging for many reasons. For instance, the body of knowledge surrounding human genomics constantly changes and evolves while scientists and researchers globally conduct research based on 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. For example, the term allele might refer to: i) an alternative form of a gene or locus[5]; 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 that are not associated with any gene. How can this concept (and its multiple underlying interpretations) be represented in a consistent way? How can scientists create knowledge based on 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 domain-specific concepts in a consistent way and, particularly, for facilitating a shared understanding of genomics, are the so-called lightweight ontologies (i.e., logical specifications typically in some form of Description Logics) and thesauruses of controlled vocabulary [1] that 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 facilitate the exchange of information [4, 5, 6], while providing a sound basis from which to make a conceptualization process explicit and facilitate the achievement of a shared understanding of a domain [7]. This will ultimately impact the effective communication among physicians, geneticists, biologists, and other researchers [8].

The objective of this research is to extend prior work on the Conceptual Schema of the Human Genome (CSHG, [9]) by making the definition of the relevant concepts of the model precise, explicit, and understandable for all. To do so, we conduct the "ontological unpacking" of the CSHG, i.e., a process that reveals implicit, relevant knowledge by transforming a traditional conceptual model into an ontologically-grounded conceptual model. We refer to "ontological" in a strong sense, because our transformation 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. As a source modeling language, we employ UML [10]; as a target, we employ the modeling language OntoUML [11], which is grounded in the Unified Foundational Ontology (UFO) [12].

Previous work on the ontological unpacking of biology-related models has been reported in [13, 14], where the method has been applied to the case of a Viral Conceptual Model [15], designed to organize the data collected about SARS-CoV-2, the virus responsible for COVID-19, as well as similar viruses. In [16], we framed our first proposal to use ontological unpacking in a conceptual model of human genomics, which we further develop and detail here.

Ontology-driven conceptual modeling has been compared to traditional conceptual modeling in [17], followed by other studies that have considered their differences in various domains [18, 19] or from a theoretical point of view [20]. We do not aim to compare different languages or paradigms, but rather, the capability of different models (an original conceptual model and its ontologically unpacked version) to completely and unambiguously convey the salient information of a complex domain. This serves the intended purpose of explaining that domain to a non-expert user who approaches it for the first time and needs a basic understanding for interacting with experts.

The manuscript is organized as follows. Section 2 introduces basic concepts on OntoUML and overviews the CSHG, used as a source conceptual model for the ontological unpacking process. Our first contribution stands in

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

reformulating a UML-based conceptual model (i.e., the CSGH) into its corresponding ontologically unpacked conceptual model – in OntoUML. This is a laborious process (described in Section 3), which requires time and modeling expertise. Then, we address the *research question* "Is it worth performing ontological unpacking on basic models (e.g., UML) to produce semantically-richer models (e.g., OntoUML) that allow a better explanation of a complex domain?". First, we answer from a theoretical point of view (Section 4); we discuss how a foundational ontology brings ontological clarity to a complex model by facilitating domain understanding and by forcing designers to unveil its complexity. Then, we answer from a practical point of view (Section 5), evaluating the usability of the proposed method with an empirical study. Overall conclusions are provided in Section 6.

2. Background

The context upon which this research is based is depicted in Figure 1: traditional conceptual modeling [21] 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 [22].

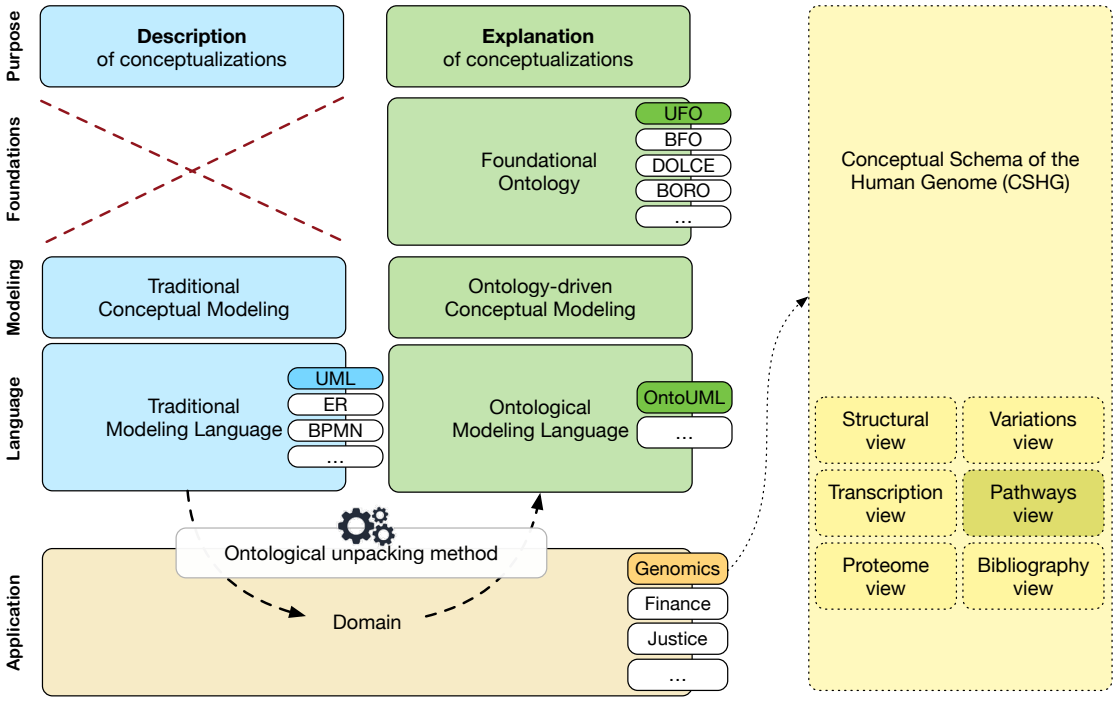


Figure 1: Schematic representation of concepts used in the manuscript. Specific instances used in the described approach (i.e., the Traditional Modeling Language UML, the Ontological Modeling Language OntoUML, the Foundational Ontology UFO, the complex Domain of Genomics, and the Pathways view of the CSHG) are highlighted with colors as they are employed to build our contribution).

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. Several languages enable us to pursue the modeling effort, among which UML [10], ER [21], and BPMNs [23] are well known. In contrast, ontology-based conceptual modeling is derived 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, method-

ologies, design patterns, and simulators) that improve the practice of conceptual modeling [12]. The two kinds of modeling have different purposes: the first aims to describe conceptualizations; the second pursues their explanation.

In contrast to traditional conceptual models, ontological modeling requires languages that are more expressive. The complete *explanation* (as opposed to *description*) of the represented domain can be achieved by grounding its modeling on a Foundational Ontology. Relevant examples include: the Basic Foundational Ontology (BFO, [24]); the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE [25]); the Business Object Reference Ontology (BORO [26]); and the Unified Foundational Ontology (UFO, [11, 12]). Few examples of ontology-driven conceptual modeling languages based on foundational ontologies are found in the literature (e.g., [27]). In this research, we employ OntoUML, which is based on UFO.

Ontological unpacking can be used to transition from an ontologically-independent representation (expressed, e.g., via UML) to an ontology-aware representation (e.g., via OntoUML). In the following, the approach will be instantiated in the context of a scientific domain. Specifically, we focus on a fragment of genomics, specifically, metabolic pathways, i.e., connected chemical reactions that occur within human cells. These are represented within a portion of the CSHG (in its Pathways view) and represented using the UML traditional modeling language. In our research we "unpack" this source model, obtaining its representation formulated with OntoUML, to reveal relevant knowledge that would otherwise remain implicit. In a conceptual model, we are attempting to faithfully represent the world as it really is. This should lead to a better understanding of the genome and better problem-solving in relation to a specific domain (here, i.e., metabolic pathways). High-quality conceptual models facilitate the creation of better information systems representations of the domain and their better design—which results in information systems implementations that tend to be more maintainable and more robust to change.

2.1. OntoUML

OntoUML [11] 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 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—see first row in Figure 2).
- UFO countenances two fundamental types of Individuals: endurants (i.e., objects and their existentially dependent reified aspects) and perdurants (i.e., events and processes). The following rows of Figure 2 summarize the main message of Individual stereotypes, as detailed next.

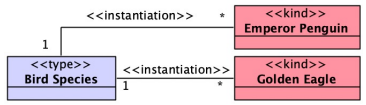
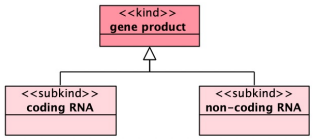
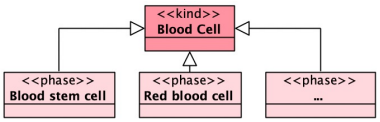
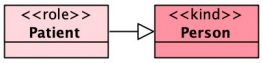
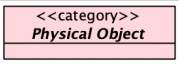
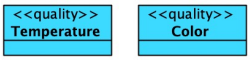
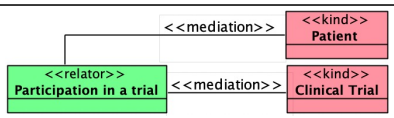
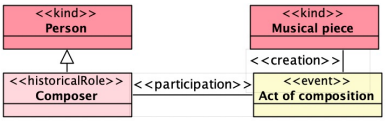
Endurants types are themselves classified based on two dimensions, i.e., sortality (identity) and rigidity. Sortals are types whose instances obey a single identity principle (i.e., are all of the same «kind»); non-sortals are types that classify instances of multiple kinds. A type is rigid if it defines essential characteristics of its instances; anti-rigid if it defines contingent characteristics for all instances. The person type is typically considered rigid (since instances of person are necessarily so), but the student type is considered anti-rigid.

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".

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. 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», which are both 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 medical treatment. Phases and roles are sortals (i.e., they classify

Figure 2: Overview of a part of OntoUML stereotypes, with their description and examples taken from the proposed ontologically unpacked model. In the OntoUML models, we employ the color coding scheme generally accepted by the community: light red is used for types whose instances are objects, green when instances are relators, yellow when instances are events, and purple when instances are higher-order types.

Stereotype	Description	Example
«type»	High-order type whose instances are themselves types.	
«kind» 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	

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 of 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).

Refactoring (non-sortal) types are rigid. They are those abstract *essential* properties common to entities of several kinds. (E.g., the category "Physical Object" represents properties of all kinds of entities that have masses, spatial extensions, etc.). 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». Qualities are aspects that can be directly associated with structured value spaces (e.g., color or temperature).

In addition to intrinsic aspects, are relational ones; i.e., entities that are existentially dependent on a multitude of

individuals, thereby 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 [28]. Events are characterized by 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 on whether they are sortals). For instance, a "composer" participated – played a historical role – within an "act of composition" event.

2.2. The Conceptual Schema of the Human Genome

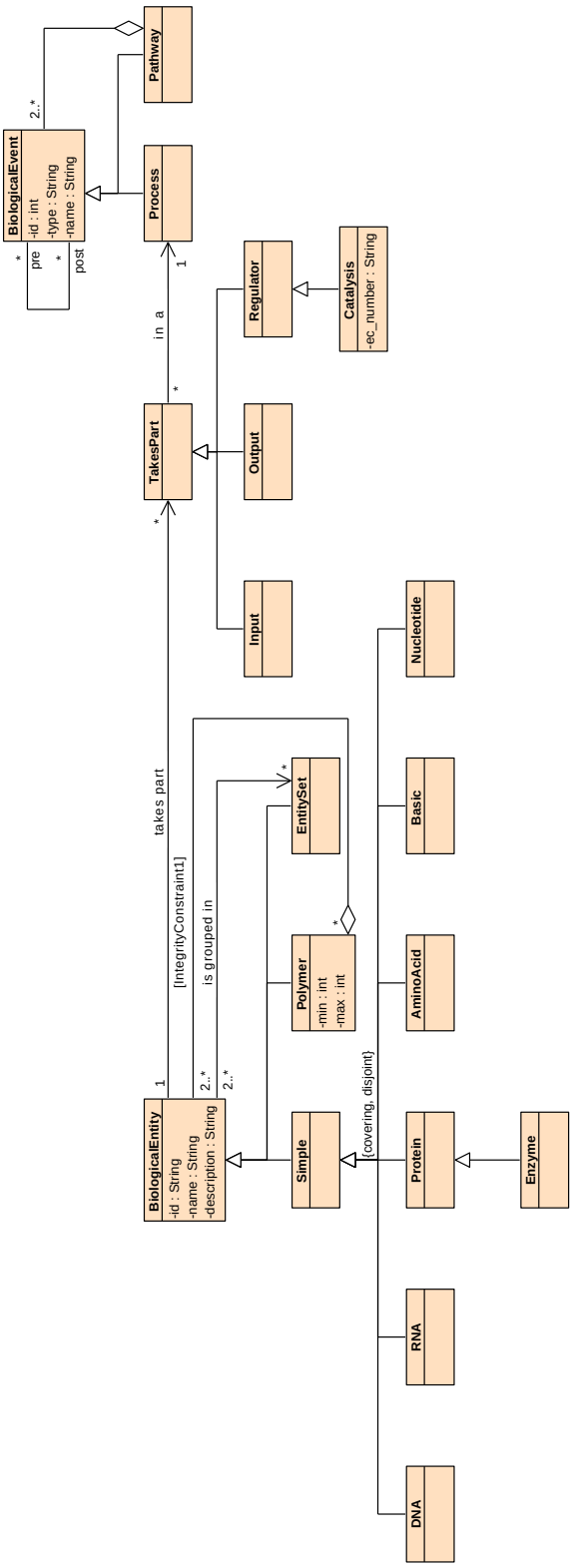
Prior research on modeling the human genome and its related body of knowledge resulted in the development of the Conceptual Schema of the Human Genome (CSHG). This conceptual schema is not limited to representing the human genome itself. It also considers the proteins and additional products generated through transcription and translation processes, how they interact within our body (i.e., the metabolome and biological pathways), and the result of such interaction (i.e., the phenotype). Our understanding of genomics evolves rapidly and so does the CSGH. 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 [29]. The model was then expanded to include the concept of phenotype and its relationships with other genomics components [30]; the second version drastically changed how the DNA sequence is represented: from a gene-centric to a chromosome-centric vision [31]. 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 expanded 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): 1) Structural view, focusing on the composition of transcribable chromosome elements (genes, exons, regulatory elements, conserved regions, etc.); 2) Variations view, identifying the types of changes that may occur in the genome; 3) Transcription view, dealing with the process of moving from DNA to RNAs; 4) Pathways view, describing the chemical reactions that explain the different molecular processes; 5) Proteome view, characterizing proteins' structure and properties; 6) Bibliography and data sources view, identifying relevant information related to sources of valid information (publications, genome data sources, etc.).

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 in this research. 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.

Consistent with the term used by geneticists, a `BiologicalEntity` class identifies any possible physical component present in our body that can play a role in a `BiologicalEvent`. In turn, a `BiologicalEvent` class represents the events that occur in our body. Biological events are recursively composed of additional events, which 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 specific type that cannot be decomposed further. A pathway is a more complex type of event that is decomposed into a specific set of events, either processes or pathways. Biological entities are of three different types: `Simple`, `Polymer`, or `EntitySet`. 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 are represented in the schema by means of the `Protein` class. Another type is the chemical compound one (e.g., water), which is represented in the model with the `Basic` class.

Figure 3: **Pathway view of the CSHG.** Orange boxes represent the classes that pertain to the Pathway view.



A polymer is an entity that concatenates two or more simple entities of the same kind a given number of times. There is an Object Constraint Language (OCL) integrity constraint (identified as "[IntegrityConstraint1]"[6] in the schema) that enforces this condition.

An entity set groups a number of biological entities of any type that can be used interchangeably because they play an equivalent role. The biological entities that belong to an entity set retain their individuality, which means they play an equivalent role but are not combined. Entity sets are used as aggregates to reduce the granularity of pathways.

Regarding biological events, a process is an atomic, specific interaction between different 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. Thus, when an entity takes part in a specific process, it assumes at least 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.

3. Ontological Unpacking

We review the original conceptualization underlying the CSHG by means of an ontological unpacking mediated by OntoUML and its underlying foundational ontology UFO. The results lead us to an improved CSHG, whose sound and precise ontological commitment fulfills the conceptual clarification that our work explores (see Figure 4). This analysis focuses on clarifying the notions of entity and event in the original model and how they relate to each other.

In the original UML class diagram of Figure 3, the concepts of `BiologicalEntity` and `BiologicalEvent` 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 (renamed to `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. This is because categories aggregate essential properties of individuals that follow different identity principles (i.e., they pertain to different kinds).

In the unpacked model, we added the stereotype «event» to the event concept (renamed to `biological_event`) to represent that they are ontological entities that unfold over time, accumulating temporal parts and mapping the world from situation to situation [28].

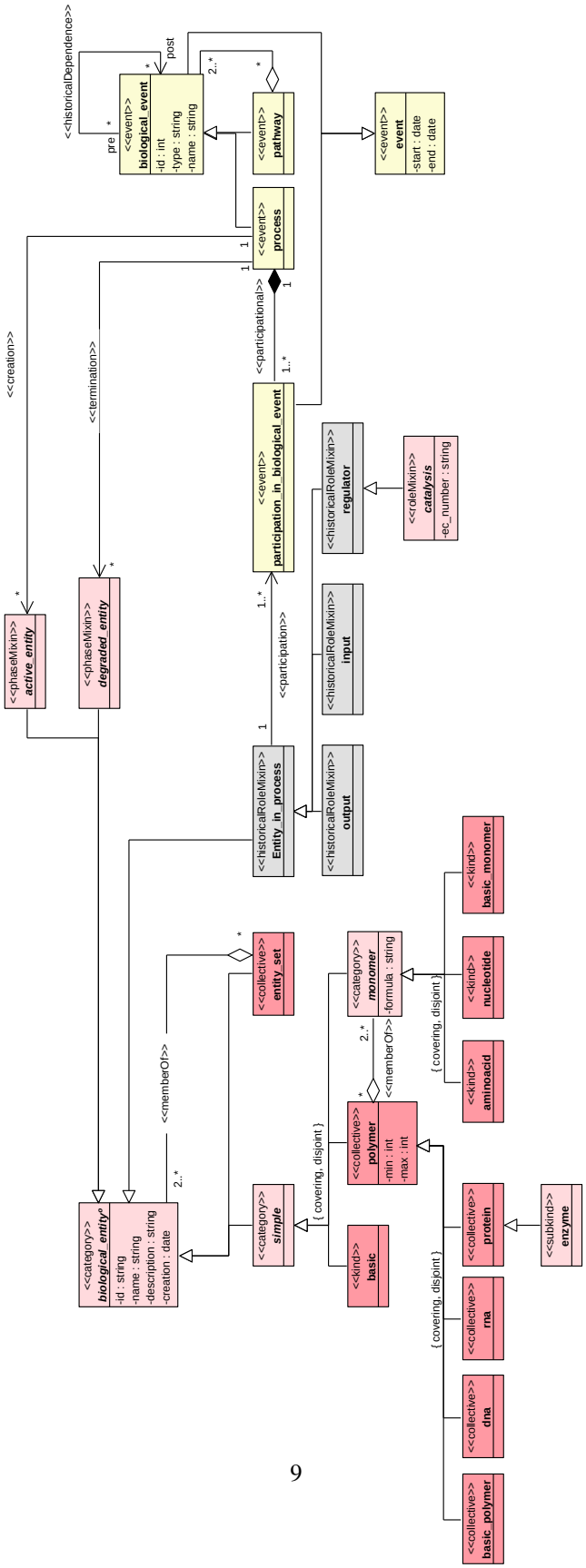
Events are of great importance in human cognition, with the need to model them explicitly. By modeling events as classes, we provide identity principles and properties, as well as rules for relating various event types. The UML version of the model was characterized by an identifier and a name. By mapping our original class, called `Event`, to its corresponding notion of `event` in OntoUML [28], we add two new attributes, *start* 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 [32]. It also distinguishes, for example, cases in which an event is eventually followed by another (i.e., after, in Allen's terms) from cases in which an event is immediately followed by another (e.g., meet).

In the original model, we had one type-reflexive relation connecting the event class with itself and with the role names "-Pre" and "-Post". However, this modeling choice 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 the «historicalDependence» stereotype [28]. This makes explicit that, if an event of type A is historically dependent on an 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.

Events can be composed of a set of other events, forming partonomies. This can be illuminated by the mereology of events underlying OntoUML [33]. Mereology accounts for two orthogonal dimensions of the decomposition of events, namely, a *structural dimension* and a *participation dimension*. For example, consider a tennis match between Novak Djokovic and Rafael Nadal. In the former dimension, this event is decomposed into sets, games, and points (each of which is an event in itself). In the latter, the match can be decomposed into Djokovic's participation (the sum of his contributions to the match, which are all events that are dependent on him) and Nadal's participation.

[6] `Polymer.entities -> forAll(e1,e2: Entity | oclType(e1).equalsIgnoreCase(oclType(e2)))`

Figure 4: **Ontologically enriched version of the CSHG.**



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* [33]) with minimum cardinality constraints on the relations. This revised part of the model is a direct instantiation of UFO's structural partonomy pattern [33].

The participation dimension is characterized by representing the role that biological entities play in processes. This was originally modeled by the `TakesPart` class in the UML schema, an entity that 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, in which biological entities have participated, as an event. In contrast to the original schema (defined using UML), 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 have mandatorily participated in an event. Historical roles explicitly describe the variety of roles that biological entities could play in the processes.

The `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 «historicalRoleMixin» (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., the participation of the ribosome and the participation of 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 unpacking 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. Since a protein is a polymer of amino acids, proteins are modeled through the `Protein` and the `Polymer` classes in the original model. 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 (i.e., as a type of simple entity)? The answer is no, because polymers are composed of monomers; one is atomic and the other is not. As a result of our analysis, we reduced the number of concepts into which a biological entity can be specialized in the unpacked version to either `simple` or `entity_set`.

We stereotyped the `simple` entity as a «category», just like the `biological_entity` class. The `polymer` class has been reevaluated as a type of `simple` element, and we created a new class, namely the `monomer`. A polymer is stereotyped as a «collective» that is composed of several instances of a single type of monomer. The monomer is a «category» that groups the set of different atomic elements that can conform polymers, and 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 the `basic_monomer`, which clusters other monomers such as glucose. Finally, the `basic` entity remains unchanged as a type of simple entity.

We stereotyped the `entity_set` entity as a «collective» to identify plural entities, which aggregate parts (i.e., members) that play the same role with respect to the whole. This definition captures 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 then becomes clearer.

4. Analysis of Ontological Unpacking Consequences

The ontological unpacking process produces an output model (expressed, in this case study, through OntoUML) from a source model (expressed, here, through UML). Here, we would like to pose a fundamental question: *Is the ontological unpacking process affecting the power of models to pursue explanations of complex domains?* In the analyzed case, the ontological unpacking 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 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.

4.1. 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 original model (Fig. 3), it is not clear how the whole and its parts are connected. That model allowed us to represent the same entity in multiple ways (e.g., a protein could be represented through both the `Protein` class and the `Polymer` class). In the unpacked version, we created the `monomer` class and connected it to the `polymer` class. We then reorganized the existing subtypes of single entities by determining whether they are polymers or monomers. This change made it easier to identify the parts (monomer) that compose the collective (polymer). It also removes the possibility of representing the same entity in multiple ways (e.g., a protein is now represented through the `protein` class, which is a subtype of polymer).

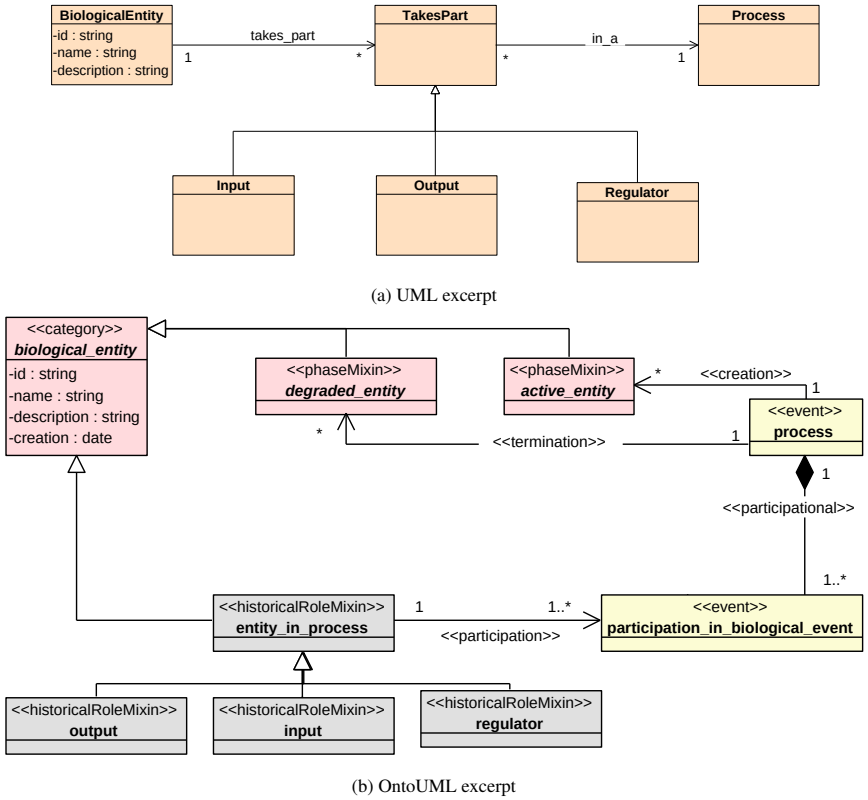
The identity and rigidity dimensions cannot be considered with the "flat" semantics of UML. UML represents objects with (e.g., «kind») and without (e.g., «category» or «roleMixin») identity principles in the same way. Similarly, for objects whose instances are rigid (e.g., «kind») or anti-rigid (e.g., «role»). Our analysis shows how these aspects affect conceptual clarity. In the unpacked version of the model (Fig. 4), we can easily identify the core components and characterize their changes that result from modifications of internal properties or external interactions.

The new characterization of biological entities also provides a clearer distinction between them. For instance, the original model characterized proteins with their own class, called `Protein`. However, this model also characterized polymers with a class. Note 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`, as a `Polymer`, or as both types? While an initial answer might be to use the `Protein` 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 unpacked model, exploiting 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 contained in the original UML model, but it requires implicit knowledge regarding genomics. The unpacked OntoUML model 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).

4.2. Changes in Biological Entities Over Time

In the original UML model (Figure 5a, which represents the relevant excerpt from the whole model presented earlier), an entity can act as an `Input`, an `Output`, or a `Regulator`. In the unpacked OntoUML model (see Figure 5b), there is an additional dimension that can 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 could not be modeled without the use of the «phase» stereotype. In the unpacked OntoUML model, this clarifies that the changes of `biological_entities` in



(a) UML excerpt

(b) OntoUML excerpt

Figure 5: Model excerpts concerning the changes in biological entities over time.

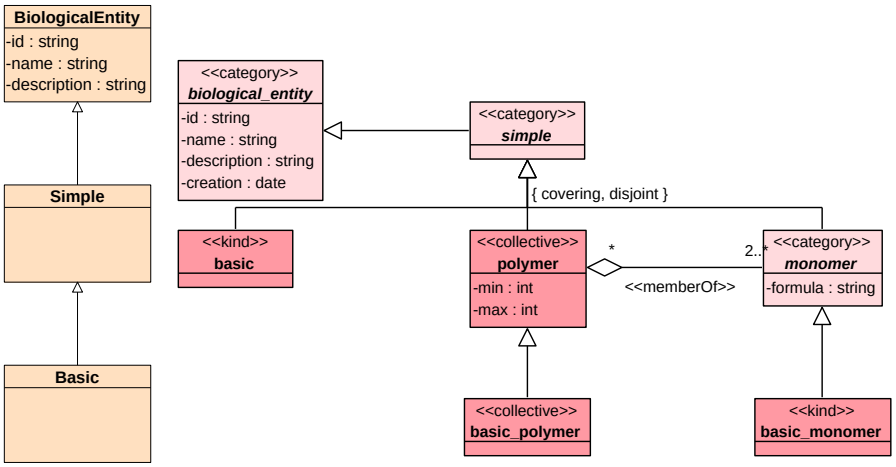
our bodies result from processes. In contrast, it is not clear how to model the degradation of entities with the original UML model.

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 during 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.

4.3. Representation of Chemical Compounds

Thousands of different chemical compounds take part in the continuous processes that occur in our bodies. In the original UML model (Fig. 6a for the relevant excerpt), they are represented with the `Basic` class, which is a type of simple entity. 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 allowed us to identify the need for modeling: i) chemical compounds that are neither polymers nor monomers; ii) chemical compounds that are polymers; and iii) the monomers of these polymers.

To increase clarity in the unpacked OntoUML model, we created two new classes (Fig. 6b): `basic_polymer` is stereotyped as a «collective» to represent chemical compounds that are polymers; `basic_monomer` is stereotyped as a «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.



(a) UML excerpt (b) OntoUML excerpt

Figure 6: Model excerpts concerning the representation of chemical compounds.

5. Empirical Evaluation

In the previous section, we concluded that the ontological unpacking process positively affects the power of explaining complex models, using specific examples from the analyzed domain. Recognizing that ontological unpacking takes time and effort, we attempt to evaluate its benefits in terms of usability (i.e., whether it provides a better explanation of a complex domain) via an empirical experiment.

Hypothesis development

To assess the usability of an ontologically unpacked model versus an original model, we used the Goal Question Metric template. The guidelines for goal definition in software engineering experiments are found in Wohlin et al. [34]). ISO 25000 [35] defines usability in terms of effectiveness, efficiency, and user 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*". Is the unpacked model achieving different effectiveness, efficiency, and user satisfaction with respect to the original model? We defined three null hypotheses to be tested throughout the experiment:

- H_{01}: The effectiveness of an ontologically unpacked model is the same as the one of its original model.
- H_{02}: The efficiency of an ontologically unpacked model is the same as the one of its original model.
- H_{03}: The user satisfaction employing an ontologically unpacked model is the same as the one of its original model.

Factor, response variables and metrics

The *factor* used in the experiment corresponds to the process used to build a conceptual model. It has two levels: the control treatment (i.e., traditional conceptual modeling, with UML notation), and the target treatment (i.e., ontology-driven conceptual modeling, with OntoUML notation). We defined one *response variable* for each null hypothesis to be tested. The first response variable is *Effectiveness*[7], measured through a questionnaire whose questions investigate the meaning of the elements represented in several parts of a conceptual model (see [37]). 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 group, we defined a metric, calculated as the sum of values associated with its answers.

[7]Effectiveness is defined by the IEEE dictionary [36] as "*the accuracy and completeness with which users achieve specified goals*."

The second response variable is *Efficiency*,[8] measured in terms of the time spent by the analyst in understanding the conceptual model, with the purpose of answering the questionnaire. For each group, we summed the time spent answering the questions.

The third response variable is *User satisfaction*,[9] measured using three metrics [38]: 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) [39]. In the MAM questionnaire, we defined six questions to measure PEOU, eight for PU, and two for ITU. Their metric is calculated as the sum of the answers for each one of them. Therefore, possible values for PEOU are within [6,30], PU [8,40], and ITU [2,10]. We defined a questionnaire for each treatment (UML original model and OntoUML unpacked model); the same questions were asked on each treatment (see questionnaire in [37]).

Subjects

The experiment was carried out with twenty subjects, selected from Computer Engineering students that are learning Model-Driven Development (MDD) in their curriculum. In their prospective career, these students will need to work in collaboration with domain experts (even of complex domains), for example, for gaining the basic understanding needed for designing information systems and databases. For this reason, they were considered good proxies of stakeholders for Information Systems design, which requires a shared understanding of a domain.

We asked each subject 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; they 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 (as junior developers).

Figure 7: Subjects' previous experience regarding genomics, class diagrams, UML language, and OntoUML language.

See Figure 7 for a visual representation of subjects' experience in the involved topics. All subjects knew about class diagrams and the UML language. The majority had previously taken or were attending classes on both (only four subjects never took a class on UML). Instead, 65% of the subjects had never heard of OntoUML before the experiment, and only two of them had studied it in their classes. Only half of the subjects took classes on genomics; three of them had never heard of genomics. For the validity of the experiment, it was important that students were not already knowledgeable in the domain chosen for the models to be examined.

[8] Efficiency is defined in the IEEE dictionary [36] as "*the degree to which a system or component performs its designated functions with minimum consumption of resources*".

[9] User satisfaction is defined in the IEEE dictionary [36] as "*freedom from discomfort, and positive attitudes towards the use of the product*".

Problem	Group	ID	Competency Questions
P1	Entities	1	Polymers are composed of other polymers.
		2	The internal structure of any polymer 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 problems

The experiment was posed for measuring how an ontologically unpacked model (expressed with OntoUML) can explain relevant concepts of a specific domain when compared to its corresponding original model (expressed with UML). During a focus group with medical doctors and geneticists, experts in genomics, we collected 26 questions (see the full list in Appendix A). These questions meet the criteria of being considered important concepts for a basic understanding of the metabolic pathways domain. Given the need to comply with the expected conditions of an experiment [34], with reasonable sample size and limited time availability, we opted for reducing the pool of questions. From the ones initially proposed, we selected the most relevant and definite 18 questions. More specifically, we removed the eight questions that were either redundant or not well-defined, or were not at the right level of detail. The remaining ones were then divided into three groups related to entities (6 questions), events (6), and interaction of entities within events (6). These questions were distributed into two different problems (P1 and P2), attempting to offer a homogeneous level of difficulty and a variety of topics. Further, we had the experts' coordinator inspect the final list and obtained confirmation on different levels. The selected questions: 1) are correct and non-redundant; and 2) allow us to cover the general knowledge of biological pathways and their components. The coordinator also confirmed specifically that the questions enabled us to capture an adequate understanding of the domain, even though the answers were binary (true/false).

Experiment design

The experiment had a within-subjects design (repeated measures), where two factors were applied to all subjects. As a block variable,[10] we considered the assigned problem (i.e., P1/P2), as we were not interested in identifying differences between problems but in analyzing if the treatment affected the results.

To prevent the order of the treatments (i.e., UML/OntoUML) or the order of the problems (i.e., P1/P2) from influencing the results, we divided the subjects into four groups. Each group represented a possible combination of problem and treatment. Groups were balanced, and subjects were randomly assigned to one group.

[10] A block variable is a variable we are not interested in studying, but we aim to ensure that it is not affecting the results.

Experiment procedure

After collecting demographic surveys from subjects, we ran two 45-minute teaching sessions on the theory and practice of, respectively, UML and OntoUML. 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) on a topic not related to genomics. Once we ascertained that the knowledge of all students was sufficient to participate in the study, we distributed the questionnaires to them 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 original UML model and its corresponding unpacked OntoUML model available in [37]. Then, they also completed one MAM questionnaire for each treatment. The operational workflows of each of the four groups are 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.

We use box-and-whisker plots to illustrate the differences regarding the treatments of the response variables. Descriptive data help graphically identify possible differences between treatments. We used a mixed model as a statistical test to identify significant differences between treatments and among replications. The mixed model can be applied under the assumption of the normality of residuals, which can be tested with the Shapiro-Wilk test applied to the residuals. This was calculated automatically during the application of the mixed model test [40]. When $p-value < 0.05$, we can reject the null hypothesis, i.e., there are significant differences in the variable. We used Cohen's d [41] to calculate the effect size in those variables with significant differences. Cohen's d is defined as the difference between two means divided by a standard deviation of the data. A value $v > 0.8$ corresponds to a large effect; $0.79 > v > 0.5$ corresponds to a moderate effect; and $0.49 > v > 0.2$ corresponds to a small effect. With G*Power [42], it can be shown that – for a repeated measurement statistical test – a sample size of 16 units for an effect size of 0.8 (large effect) is big enough to achieve a power of 80%. Thus, the 20 sample units (i.e., students) considered in our experiment exceeded the minimum requirement to conduct the statistical analysis.

5.1. Results

5.1.1. Effectiveness and Efficiency

The Effectiveness and Efficiency variables have been measured separately for the three groups of questions. Figure 8 shows box and whiskers plots regarding Effectiveness (panels A, B, and C) and Efficiency (panels D, E, and F). Panels A and D show the box plots for the entities group, panels B and E for the events group, and panels C and F for the interaction between entities and events. The lines that connect treatments' blue boxes represent the averages. In panels A and B, the median, first quartile, and third quartile show that the unpacked model (in OntoUML) yields better effectiveness than the original model (in UML). However, Panel C shows that the median, first, and third quartile is the same for both treatments. In panels D–F, the median, first quartile, and third quartile are higher for the unpacked model, showing that the original model yields better efficiency than the unpacked one.

Table 3 shows the statistical analysis of effectiveness (left) and efficiency (right) for their different metrics. We detail the results of applying the Mixed Model to the data for entities, events, and entities in events.

Regarding effectiveness, 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. No significant differences in the Method*Problem are observed for these metrics (see Interaction column in Table 3). This means that the problems (i.e., P1/P2) used in the experiment do not affect the results. We can reject H_{01} for the Entities and Events metrics, i.e., the effectiveness of an ontologically unpacked model is higher than the one of its original model.

Regarding efficiency, all metrics yield significant results (p-values < .05). Thus, answering problems using an ontologically unpacked model in OntoUML requires significantly more time than using the original model in UML. Also here, no significant differences in the Method*Problem were observed for these metrics; thus, the choice of the problem is not affecting the results. We can reject H_{02} for all metrics; i.e., the efficiency of the original model is better than the one of the ontologically unpacked model.

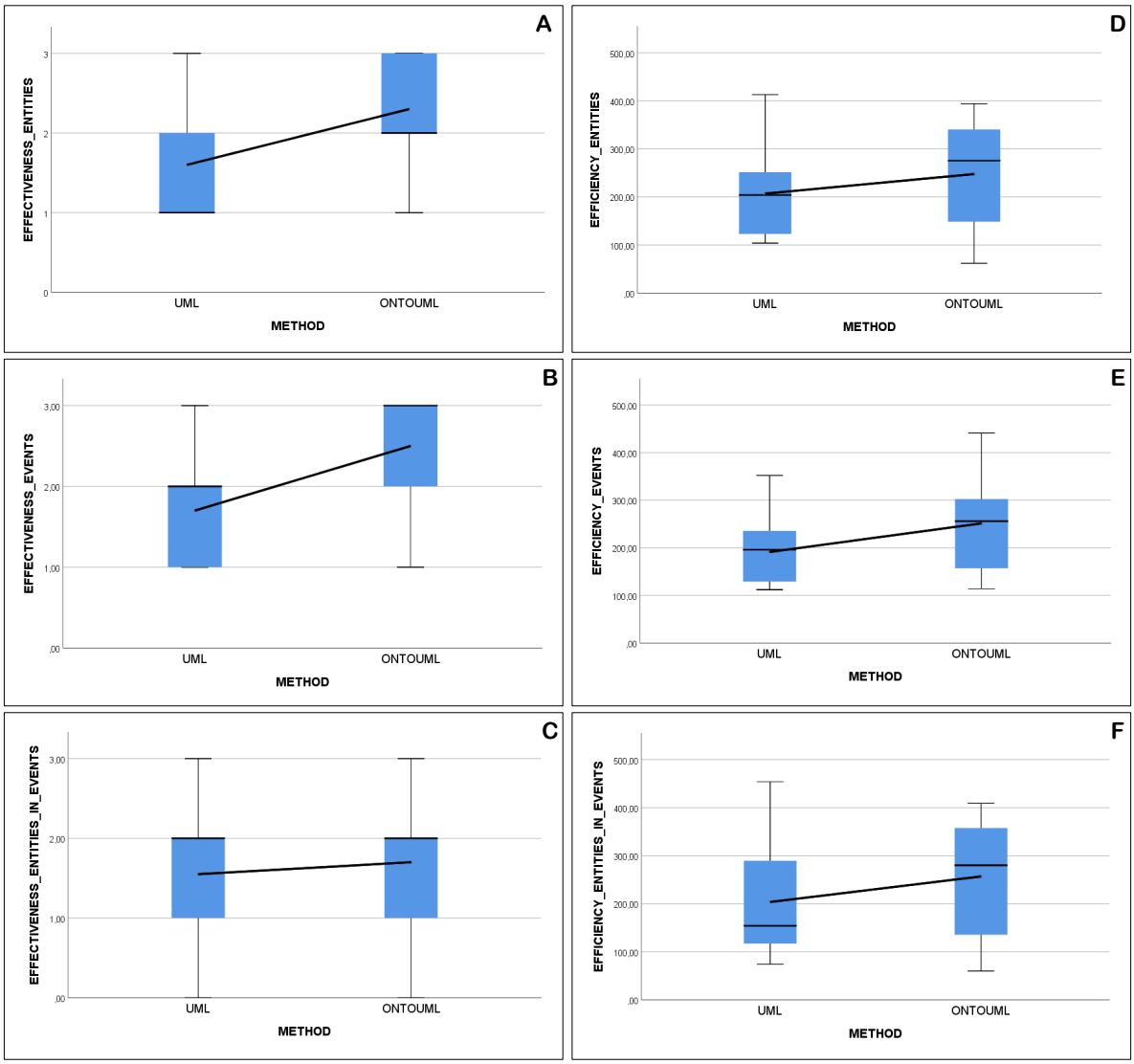


Figure 8: Box and whiskers plots for the effectiveness of entities (Panel A); the effectiveness of events (Panel B); the effectiveness of the interaction between entities and events (Panel C); the efficiency of entities (Panel D); the efficiency of events (Panel E); the efficiency of the interaction between entities and events (Panel F).

5.1.2. User satisfaction

Last, we analyze the results for the variable *User satisfaction*, by separately considering its three metrics: perceived ease of use, perceived usefulness, and intention to use. Figure 9A shows the box plot for perceived usefulness. Median, first, and third quartiles show higher satisfaction using the original UML model. This pattern is also repeated for perceived usefulness (Figure 9B and intention to use (Figure 9C).

Table 3: Data analysis results for effectiveness and efficiency metrics.

	Effectiveness				Efficiency			
	Treatment	**Interaction**	**Mean**	**Effect Size**	**Treatment**	**Interaction**	**Mean**	**Effect Size**
ENTITIES	****.001**	.112	UML: 1.60 OntoUML: 2.30	**.98**	****.006**	.165	UML: 206.95 OntoUML: 247.65	.4
EVENTS	****.001**	.388	UML: 1.70 OntoUML: 2.50	**1.2**	****.000**	.731	UML: 191.25 OntoUML: 251.4	.71
ENTITIES IN EVENTS	0.587	.285	UML: 1.55 OntoUML: 1.70	-	****.001**	.468	UML: 203.65 OntoUML: 256.85	.44

Table 4 shows the three metrics running the Mixed Model, all with significant results (p-value <.05), yielding UML a better average rather than OntoUML. We can then reject H_{03} for all metrics, which indicates that the satisfaction of users working with the original UML model was significantly better than the one of users working with the ontologically unpacked OntoUML model.

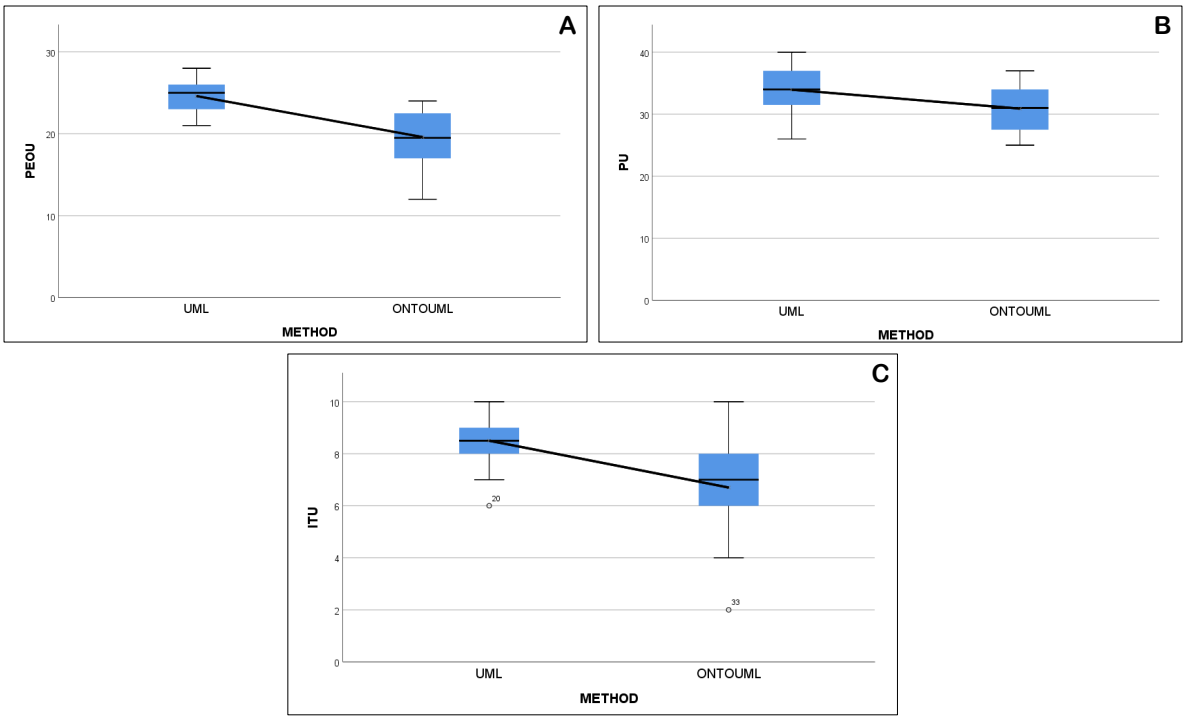


Figure 9: Box and whiskers plot for perceived ease of use (Panel A); perceived usefulness (Panel B); and intention to use (Panel C).

5.2. Discussion

For *Effectiveness* (H_{01}), the empirical analysis allowed us to conclude that the unpacked model in 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: i) Entity-related questions were answered more successfully with the unpacked model. This is likely because UFO contains stereotypes that helped clarify important principles, such as rigidity. ii) Events-related questions were also answered more successfully with the unpacked model, 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

Table 4: Data analysis results for satisfaction metrics.

	User satisfaction			
	Treatment	**Interaction**	**Mean**	**Effect Size**
PEOU	****.005**	.843	UML: 8.50 OntoUML: 6.70	**1.1**
PU	****.003**	.923	UML: 33.95 OntoUML: 30.90	.78
ITU	****.005**	.843	UML: 8.50 OntoUML: 6.70	**1.1**

entities were answered more successfully with the unpacked model by a very small fraction. A number of interesting considerations arise:

- Conceptual modeling aims to make implicit concepts explicit. From a biological point of view, events are clearly limited in time. However, in the original model in UML (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 a higher percentage of participants to respond correctly.

- The original model provides a simple explanation of the participation of entities in the processes (Figure 3), whereas the unpacked model (Figure 4) provides a more complex and detailed explanation. Note that using an ontologically-grounded language such as OntoUML has allowed us to analyze the *mereology of events*. In particular, Q6 was answered with a higher score using the unpacked model (70%) instead of the original model (40%), likely because the latter 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 original model (80%) rather than with the unpacked model (%20). In this case, respondents were probably confused by the complexity of the representation, which should be the object of further study and evaluation.

- OntoUML expresses the «phase» stereotype, exploiting the principle of rigidity [11], which clarifies the fact that chemical compounds and biological-related substances are created and destroyed as a result of chemical reactions. Q16, regarding this aspect, showed a significant difference between the two models (OntoUML 90% vs. UML 30%), demonstrating the higher capability of the unpacked model of explaining such a principle.

Note that the experiment did not intend to compare the two languages used for representation. Instead, it served the purpose of understanding if the additional semantics brought by OntoUML, in the context of the analyzed complex domain, is effective. This semantics could have made the model unnecessarily more complex and confused subjects.

The *Efficiency* assessment was measured through H_{02}; the null hypothesis could be rejected for all groups, as responding to questionnaires using the unpacked model required longer response times. This was likely due to the complexity of the OntoUML language and the 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. In contrast with this, the evaluation showed that the unpacked model required more time for participants to be able to answer questions based on it.

The *User satisfaction* assessment, tested through H_{03}, showed that, for all groups, the use of the unpacked model was, in general, less appreciated by users. They were hesitant to learn and use a novel modeling language, especially a complex one, in a short amount of time.

However, crucially, the results indicate that performances, in terms of effectiveness, were significantly better using the unpacked model. As shown in other research [17, 18], subjects generally need more time to properly understand the paradigm of ontological conceptual modeling and, specifically, the intricacies of the OntoUML language. Therefore, further experimentation on PEOU, PU, and ITU should be repeated with subjects that have received longer training.

In summary, the practical adoption of the ontological unpacking method is currently hindered by the long learning curve of formalism on the part of users. It is apparent that a previous background in OntoUML would greatly facilitate 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 the 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., with respect to conclusion validity, internal validity, construct validity, and external validity), as defined for quasi-experimentation by Cook and Campbell [43], in line with [44, 45]. Threats to *conclusion validity* [34] 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 [42] to estimate the minimum sample size needed for achieving statistical significance;

ii) the *reliability of measures*, which was mitigated by asking both domain experts and non-experts to double-check the list of questions for proper wording and interpretation;

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 a homogenous knowledge level on Class Diagrams and without previous knowledge of OntoUML and genomics (see Figure 7). 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* [34] affect the experimental factor (i.e., the modeling formalism) with respect to causality. Several threats can be mitigated by performing a multiple-group experiment (vs. a 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 (i.e., original model in UML and unpacked model in OntoUML) to two similar problems (P1 and P2) in a different order (see Table 2). No interaction between the groups was allowed.

Threats to *construct validity* [34] 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 instructional classes on the involved treatments of the same duration – properly 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. Participants, possibly developing *evaluation apprehension*, were reassured that no marks would be derived from the experiment. To reduce any *experimenter expectantions* that could bias the results, the raised questions were prepared by external domain experts.

Finally, threats to *external validity* [34], 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 [46].

6. Conclusions

The modeling of human genomics information 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 genomics with models that support 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 unpacked using 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 [47] specification for the model. These 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 investigate how to add the OntoUML notions of 'situation' and 'disposition'. Situations represent transformations from portions of reality to other ones, 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. Such additions should encourage genomics researchers to adopt the proposed models. Moreover, further ontological unpacking will be applied to other conceptual views of the Conceptual Schema of the Human Genome, focusing on the structural, variations, transcription, and proteome aspects.

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

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 the processing of heterogeneous genomics data.

Appendix A. Supplementary Data

Table A.5: The set of 26 questions collected during a focus group with different domain experts.

Sentence	True/False
A complex entity only exists in the context of an event.	F
A DNA polymer cannot take the role of input in a process.	F
A process can be decomposed into other events.	F
A protein can take the role of input in different processes.	T
A protein can take the roles of input, output, and regulator in different processes.	T
A protein can take the roles of input, output, and regulator in the same process.	T
Biological entities can be created and destroyed as a result of a process.	T
Biological entities can participate in multiple processes.	T
Biological entities can take part in pathways.	F
Biological entities can take part in processes.	T
Enzymes are a type of protein.	T
Events occur in a specific time interval.	T
Every biological entity must participate in at least one process.	F
Every enzyme is a polymer.	T
Every event must have a preceding event.	T
Pathways are decomposed in at least two events.	T
Pathways can be composed of other pathways.	T
Pathways can be decomposed only into processes.	F
Pathways must be composed of other pathways.	F
Polymers are composed of other polymers.	F
Processes are limited in time.	T
Some basic biological entities can be polymers also.	T
Some polymers are composed of amino acids.	T
Some polymers are composed of nucleotides.	T

Table A.5 continued from previous page

Sentence	True/False
The internal structure of basic biological entities and polymers is the same.	F
The internal structure of any polymer is homogeneous.	T

References

[1] B. Smith, J. Williams, S.-K. Steffen, The ontology of the gene ontology, in: AMIA Annual Symposium Proceedings, Vol. 2003, American Medical Informatics Association, 2003, p. 609.

[2] P. Gaudet, C. Dessimoz, Gene ontology: Pitfalls, biases, and remedies, in: C. Dessimoz, N. Škunca (Eds.), The Gene Ontology Handbook, Methods in Molecular Biology, Springer, 2017, pp. 189–205.

[3] S. Schulz, H. Stenzhorn, M. Boeker, B. Smith, Strengths and limitations of formal ontologies in the biomedical domain, Revista electronica de comunicacao, informacao & inovacao em saude: RECIIS 3 (1) (2009) 31.

[4] A. Olivé, Conceptual modeling of information systems, Springer Science & Business Media, Berlin Heidelberg, 2007.

[5] O. Pastor, J. Gómez, E. Insfrán, V. Pelechano, The OO-method approach for information systems modeling: from object-oriented conceptual modeling to automated programming, Information Systems 26 (7) (2001) 507–534.

[6] O. Pastor, J. C. Molina, Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling, Springer Science & Business Media, Berlin Heidelberg, 2007.

[7] O. Pastor, Conceptual modeling of life: beyond the homo sapiens, in: International Conference on Conceptual Modeling, Springer, 2016, pp. 18–31.

[8] Ó. Pastor, A. P. León, J. F. R. Reyes, A. S. García, J. C. R. Casamayor, Using conceptual modeling to improve genome data management, Briefings in Bioinformatics 22 (1) (2021) 45–54.

[9] A. García, A. L. Palacio, J. F. R. Román, J. C. Casamayor, O. Pastor, Towards the understanding of the human genome: a holistic conceptual modeling approach, IEEE Access 8 (2020) 197111–197123.

[10] G. Booch, I. Jacobson, J. Rumbaugh, The unified modeling language, Unix Review 14 (13) (1996) 5.

[11] G. Guizzardi, Ontological foundations for structural conceptual models, CTIT, Centre for Telematics and Information Technology, Twente, Netherlands, 2005.

[12] G. Guizzardi, G. Wagner, J. P. A. Almeida, R. S. Guizzardi, Towards ontological foundations for conceptual modeling: The unified foundational ontology (ufo) story, Applied ontology 10 (3-4) (2015) 259–271.

[13] G. Guizzardi, A. Bernasconi, O. Pastor, V. C. Storey, Ontological unpacking as explanation: The case of the viral conceptual model, in: International Conference on Conceptual Modeling, Springer, 2021, pp. 356–366.

[14] A. Bernasconi, G. Guizzardi, O. Pastor, V. C. Storey, Semantic interoperability: ontological unpacking of a viral conceptual model, BMC bioinformatics 23 (11) (2022) 491.

[15] A. Bernasconi, A. Canakoglu, P. Pinoli, S. Ceri, Empowering virus sequence research through conceptual modeling, in: International Conference on Conceptual Modeling, Springer, 2020, pp. 388–402.

[16] A. García S, G. Guizzardi, O. Pastor, V. C. Storey, A. Bernasconi, An ontological characterization of a conceptual model of the human genome, in: J. De Weerdt, A. Polyvyanyy (Eds.), Intelligent Information Systems, Springer International Publishing, Cham, 2022, pp. 27–35.

[17] M. Verdonck, F. Gailly, R. Pergl, G. Guizzardi, B. Martins, O. Pastor, Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study, Information Systems 81 (2019) 92–103.

[18] M. Verdonck, F. Gailly, S. de Cesare, Comprehending 3d and 4d ontology-driven conceptual models: An empirical study, Information Systems 93 (2020) 101568.

[19] D. Kalibatiene, J. Miliauskaitė, A systematic mapping with bibliometric analysis on information systems using ontology and fuzzy logic, Applied Sciences 11 (7) (2021) 3003.

[20] C. M. Keet, Z. Khan, Foundational ontologies: From theory to practice and back, Journal of Knowledge Structures and Systems 3 (1).

[21] P. P.-S. Chen, The entity-relationship model—toward a unified view of data, ACM Transactions on Database Systems (TODS) 1 (1) (1976) 9–36.

[22] J. Mylopoulos, Conceptual modelling and telos. conceptual modelling, databases, and case: An integrated view of information system development, New York: John Wiley & Sons 49 (1992) 68.

[23] R. Flowers, C. Edeki, Business process modeling notation, International Journal of Computer Science and Mobile Computing 2 (3) (2013) 35–40.

[24] R. Arp, B. Smith, A. D. Spear, Building ontologies with basic formal ontology, Mit Press, Cambridge, 2015.

[25] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, L. Schneider, The WonderWeb Library of Foundational Ontologies and the DOLCE ontology, WonderWeb Deliverable D18, final report (vr. 1.0, 31-12-2003).

[26] C. Partridge, Business objects, Re-enginnering for re-use (2nd ed.). UK: The BORO Centre.

[27] R. C. Waldemarin, C. R. de Farias, OBO to UML: Support for the development of conceptual models in the biomedical domain, Journal of Biomedical Informatics 80 (2018) 14–25.

[28] J. P. A. Almeida, R. A. Falbo, G. Guizzardi, Events as entities in ontology-driven conceptual modeling, in: Conceptual Modeling: 38th International Conference, ER 2019, Salvador, Brazil, November 4–7, 2019, Proceedings 38, Springer, 2019, pp. 469–483.

[29] O. Pastor, A. M. Levin, M. Celma, J. C. Casamayor, A. Virrueta, L. E. Eraso, Model-based engineering applied to the interpretation of the human genome, in: The Evolution of Conceptual Modeling: From a Historical Perspective towards the Future of Conceptual Modeling, Vol. 6520, Springer, 2011, pp. 306–330.

[30] J. F. Reyes Román, O. Pastor, J. C. Casamayor, F. Valverde, Applying conceptual modeling to better understand the human genome, in: Conceptual Modeling: 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings 35, Springer, 2016, pp. 404–412.
[31] J. F. Reyes Román, 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, accepted: 2018-03-22 (Mar. 2018). doi:10.4995/Thesis/10251/99565.
[32] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM 26 (11) (1983) 832–843.
[33] G. Guizzardi, G. Wagner, R. de Almeida Falbo, R. S. Guizzardi, J. P. A. Almeida, Towards ontological foundations for the conceptual modeling of events, in: International Conference on Conceptual Modeling, Springer, 2013, pp. 327–341.
[34] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Springer, Berlin Heidelberg, 2012.
[35] ISO/IEC, Iso/iec 25000 - software engineering - software product quality requirements and evaluation (square) - guide to square, `https://www.iso.org/standard/64764.html`, last accessed: April 12, 2023 (2014).
[36] IEEE, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, 1991. doi:10.1109/IEEESTD.1991.106963.
[37] A. García S., A. Bernasconi, UML vs OntoUML analysis results [Data set], last accessed: April 12, 2023. doi:10.5281/zenodo.6616114.
[38] F. D. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology, MIS Q. 13 (3) (1989) 319–340.
[39] D. L. Moody, The method evaluation model: a theoretical model for validating information systems design methods (2003).
[40] L. S. Meyers, G. Gamst, A. Guarino, Applied multivariate research: design and interpretation, SAGE Publications, Thousand Oaks, California, 2006.
[41] L. Cohen, Statistical power analysis for the behavioral sciences, 2nd Edition, Lawrence Earlbaum Associates, New York, New York, 1988.
[42] F. Faul, E. Erdfelder, A.-G. Lang, A. Buchner, G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences, Behavior Research Methods 39 (2) (2007) 175–191.
[43] T. D. Cook, D. T. Campbell, A. Day, Quasi-experimentation: Design & analysis issues for field settings, Vol. 351, Houghton Mifflin Boston, 1979.
[44] J. Parsons, L. Cole, What do the pictures mean? guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques, Data & Knowledge Engineering 55 (3) (2005) 327–342.
[45] A. Burton-Jones, Y. Wand, R. Weber, Guidelines for empirical evaluations of conceptual modeling grammars, Journal of the Association for Information Systems 10 (6) (2009) 1.
[46] D. Falessi, N. Juristo, C. Wohlin, B. Turhan, J. Münch, A. Jedlitschka, M. Oivo, Empirical software engineering experts on the use of students and professionals in experiments, Empirical Software Engineering 23 (1) (2018) 452–489.
[47] W3C OWL Working Group, Web ontology language (owl), `https://www.w3.org/OWL/`, last accessed: April 12, 2023.