

FedCSIS_2023_paper_7814.pdf

A Domain and Semantic Modeling Approach for the Development of Interactive Systems

Saulo Silva
0000-0002-0675-7173

University of Minho
ALGORITMI R&D Centre / LASI, Braga,
Portugal
Email: saulo.silva@ifg.edu.br

Orlando Belo
0000-0003-2157-8891

University of Minho
ALGORITMI R&D Centre / LASI, Braga,
Portugal
Email: obelo@di.uminho.pt

Abstract — Developing interactive systems requires tools and knowledge from diverse domains. The combination of Software Engineering with Human Factors and Ergonomics aspects assists designers and engineers in designing error free, safe systems for human operators. This paper explores the use of multidimensional knowledge representation as means to narrow the gap between design and implementation of interactive systems. We will present and discuss four distinct models, User Model, Device Model, Environment Model and Task Model, for delivering heavyweight ontologies, by the means of Ontology-Driven Development (ODD). This might enable the development of software artifacts, with potential of improving system specification, information sharing and reusability. We will discuss the theoretical approach as well as present a literature revision in the topic along with a taxonomy, enabling the description of user, device, task and environment's characteristics. Finally, we present and scrutinize the development of the ontology.

Index Terms — Human-Computer Interaction, Software Artifacts, Model-Driven Development, Ontology-Driven Development, Interactive Systems.

I. INTRODUCTION

Interaction with information systems imposes the existence of a crucial element: the user interface. It is the part of the system that humans use for contacting and accessing physically, perceptually and conceptually services provided by the information system [1]. Usually, user interfaces that provide graphical elements for controlling the system – e.g. buttons, screens, text boxes, and sliders, among others – are recognized as Graphical User Interfaces (GUI). User Interfaces seek providing system representation for human operators, exposing the control logic behind system functioning. Humans make use of the user interface for carrying out their goals with the information system.

The complexity that arises from the process of interaction between humans and machines has the potential of compromising such goal achievement. For computer systems with critical functions, such as medical devices, energy or aeronautical systems, referred to as interactive critical systems or high assurance systems, problems in user interface design may cause losses of different nature, such as financial, environmental or social. We use to characterize these losses as accidents [2]. Thus, the development of interactive systems

requires tools and knowledge from several domains, for tackling challenges in the development of safe user interfaces. The combination of Software Engineering, Human Factors and Ergonomics aspects is essential, not only for delivering adequate user experience but also for assisting designers and engineers in the design of error free, safe systems for human operators. User interface experts use information from those domains for perceiving the most relevant aspects for an ideal system use, providing the best solutions in terms of computer controls suitable for human expectations, and consequently averting gaps that could induce interaction errors. In this context, modeling can play an important role in assisting designers during user interface development.

Model-Driven Development (MDD) is a software development paradigm that makes use of information present in models for reasoning about problems and solutions regarding aspects implemented in software artifacts. Hence, user, device, task and environment modeling, for instance, might provide important insights about the description of potential system user, the computational system in use, the tasks that one must accomplish as well as the environment where the interaction occurs, respectively.

Another approach for information organization relies on knowledge representation or on ontologies, for representing explicit specifications of conceptualization [3]. For [4], ontologies might have different roles in Software Engineering (SE) field. They may be a conceptual basis for software components specification and development. Ontology-Driven Development (ODD) is a software engineering vision considering the need of semantic constraints for supporting development processes, “through which software becomes a direct projection of a semantic definition” [5]. Those constraints are valuable assets for requirement analysis activities, with the potential of reducing costs and assuring the ontological adequacy of the user interface of an information system.

Despite the specificities of ODD and MDD approaches, ontologies may be seen as models. ODD might be considered as a particular type of MDD and might assist in describing problem domain at development time, with potential of reducing the cost of conceptual analysis or assisting in the constraints/requirements phase by providing relations be-

tween knowledge elements extracted from application domains.

By providing human factors and ergonomics knowledge in the context of user interface development processes, this paper presents a work that seeks narrowing the gap between design and implementation of user interfaces. Our strategy is discussing the necessity and usefulness of human, device, task and environment knowledge in user interfaces development, describing a multidimensional knowledge representation model related with these (abstract) models, and presenting an ontology approach based on multidimensional models. We argue that these models have the potential of assisting designers and engineers in development phases as early as requirements and specification, by replacing less expressive models. The remaining of this paper is organized as follows. Section 2 presents the theoretical approach of knowledge development. Section 3, includes a literature revision in the topic of ontology development, reveals the interaction ontology we developed along with the taxonomy enabling the description of model's characteristics, and discusses how interactive systems development can benefit from using ontologies. Section 4 presents as well as analyzes the ontological development approach. Finally, Section 5, presents some remarks and conclusions, and reveals some research lines for future work.

II. RELATED WORK

According with [6], a (abstract) human model can be considered as a set of human parameters, represented by variables, which can be employed for describing users of a product. The model is described using a machine and human-readable format and is relevant for a number of disciplines, because it provides human factors information about users that can be considered during system design. Due to this, a number of standards have been created to provide user-modeling guidelines. The parameters considered in such models seek assisting systems in dealing with inter-user variance. According with [7], it refers to the variance of user parameters that must be recognized in system specification, which requires that the design cope with different types of users in the system users. The intra-user variance refers to monitoring internal user knowledge that is necessary for performing the task. User knowledge is especially relevant in the design of interactive systems. It supports different types of analysis that have the potential of preventing unwanted user interface interactions.

Device models are also relevant in designing interactive systems, since it demands rigorous design of user interface structure and behavior for avoiding hazards. Device information can provide valuable understanding of user interface parameters, which can be considered in the analysis. This is especially relevant for interactive systems with critical functions, or high assurance systems. Device modeling allows for investigating user interface structural and behavioral parameters that are presented as actions that users are allowed to perform for achieving their goals and keeping per-

formance. The interface structure is useful for reasoning about the user interface adequate working, in terms of structure's individual components (for instance, buttons or displays). It is motivated by the need of reasoning about the quality of this structure [8]. For instance, device based user interface individual components, such as buttons, when pressed, causes the activation or deactivation of internal system processes, and displays which are employed for monitoring changes in internal variables of systems. User interface behavior is useful for reasoning about the adequate user interface working behavior, resulting in the individual components coordination of such systems, being motivated by the need for reasoning about the quality of this behavior – sets of user interface components, such as panels, which are a collection of components, like displays or buttons, having different working processes intending the control of different output signals.

Environment Model has undeniable relevance when investigating system use. Within the Computer Science, Ergonomics and Human Computer Interaction communities, a number of factors related with the environment dimension such as color, air quality, light, noise and temperature, have physiological and psychological effects in humans, wherefore might affect their performance [9] (in [10]). Human-Computer Interaction (HCI) theories hold that humans perceive the environment by making use of input/output channels using the human senses (hearing, touch or sight). Next, according with the Human Processing Model theory [11], the environmental information is encoded in the brain, in which later, it is to be decoded or interpreted on demand. These processes contribute for the creation of representations from the environment in the human mind. Representations affect how she or he perceives the environment, both in terms of reason and emotion, i.e., it has both philosophical and psychological impact on humans. [12] discusses the psychological nature of representation and perception, presenting himself an alternative model for representation.

During the development of interactive systems, designers and engineers must understand what are the tasks the user must execute to achieve the desired outcome. The study of tasks intersect a number of fields, due to its transversal nature and relation with the world of work, which explains the existence of different definitions in the literature, suiting different domains. In despite of they converging to similar meaning, it is relevant to use a definition in the context of HCI and SE domains. An attempt to unify the definition is given by [13], which theorized that “a task is a goal combined with the ordered sets of tasks and actions that would satisfy it in the appropriate contexts”.

Task modelling is considered as a valuable resource intending to assist researchers from different application domains in understanding how task parameters might impact domain performance. Typically employed in the early phases of system design, it represents the relations among the tasks describing the actions carried out by the users with the system. For this reason, it has been one of the central models

in different model-based and model-driven development approaches [14].

The relevance of human factors modeling in the development of interactive systems is undeniable. It has the potential of highlighting crucial and critical aspects that might contribute for positive outcomes such as user satisfaction, improving efficiency and productivity during the interaction process, along with decreasing costs. Semantic modeling and analysis can play an important role in this demand, as models with semantic constraints have the potential in assisting designers and system engineers during the requirements phase of interaction system development. Knowledge Engineering is a discipline closely related with SE that involves methods for knowledge representation [15]. According with [16], knowledge is considered as interrelations between entities and relationships, highlighting that “relationships between entities provide meaning, at the same time that entities derive their meaning from their relationships”. Knowledge representation is then a research field that makes use of modeling activity for creating knowledge bases, useful for representing conceptualizations that are assumed existing in a certain domain of interest, along with the relations between them. From the knowledge viewpoint, [17] defines ontology as the most useful organization of knowledge about a particular domain, evolving from the need of satisfying formal postulates of signification. This is achieved by the use of object-attribute-value (O-A-V) triplets, which in ontology theory represents facts about objects and their attributes [18].

The literature covers multiple strategies on ontology development, according with [19] and [20]. It is important highlighting their main differences, as well agreeing on at least one. Ontology development approaches differs on their formalism level. Some approaches base their methodology on elaborating “tasks” or “principles” for ontology development (for instance, the methodology proposed by [21]. Other strategies focus on narrowing the distance from ontology development to other engineering disciplines, incorporating a defined life cycle for it (for instance, the work of [22], in establishing well-defined and iterative phases for ontology development). According with [5] states that more importantly is agreeing on iterative and structured process, which expresses an ontology life cycle, with well-defined processes for each activity for building and maintaining the ontology, such as:

- Domain determination – consider the domain whose syntactic and semantic knowledge are modeled into a knowledge model.
- Terms enumeration – enumerate important expressions, which describe the domain or scope and can contain underlying concepts terms, properties related with those terms, and others [21].
- Definition of class and hierarchy – develop the first idea of a class and its subclasses (i.e., a hierarchy) by examining the objects in previously defined concepts. The approaches might be top-down (classes are first

defined and then specialized in its subclasses), bottom-up (most relevant subclasses are enumerated and then grouped in different classes) or a combination of both with no loss for the result.

- Class properties definition – also referred to as slots, define the internal structure of the class. All subsequent classes inherit the slots of their super classes.
- Domain and range of properties – define the requirements for the information slots hold, i.e., its allowed values (also referred to as facets definition). A facet describes values characteristics such as type, domain and cardinality and therefore has the potential of enforcing constraints for slots-values. For instance, slots-value *type* describes what types of values are allowed in the slot. Accepted types range from *String*, *Number*, *Boolean*, *Enumerated* or *Instance*. Slots-value *domain* refers to the classes describing the slot, which is hence referred to as the range of a slot. Slots-value *cardinality* is a property that refers to the quantity of values assigned to a slot. Some slot values require a single cardinality for correctly describing certain aspects of the reality (e.g., an animal has one body). Multiple-cardinality describes any number of values (e.g., the number of species of the animal kingdom).
- Properties instances – create individual instance(s) by i) choosing a class, ii) creating individual instance of that class, and iii) filling all the defined slots of that class [21].
- Relations Definition – uncover associations expressed between the characteristics and groups by identifying and characterizing relations (also referred to as *connections* or associations). According with [23] explains these identifications could allow the relational knowledge being utilized in a near future for making useful inferences. For instance, the relation *is part of*, holds an association between two entities such as *software_button* and *user_interface*, in which it is built in.
- Data properties – its goal is twofold. They are used for connecting classes with specific data values, supporting the parameters that compose the models. Data properties are also used for restricting classes named *Defined Classes* under necessary and sufficient conditions.

We can find other examples of works in the literature related with ontologies in the context of SE life cycle, such as the work of [23] who proposed the IDEF5 ontology development method for promoting knowledge sharing and terminology standardization. The authors of [24] described ontologies classifications and proposed a taxonomy for assisting Software Engineering and Technology (SET) experts for understanding the linking between ontologies with some software development aspects.

TABLE I.
EXCERPT OF USER CHARACTERISTICS AND GROUPS

Group name	Characteristic name
Demographic data	Age, Gender
Anthropometric data	Weight, Stature
Interaction related states	Perceived stress, Stress and reaction time, Perceived Fatigue, Perceived Attitude towards computer, Motivation
Hearing parameters	Hearing

TABLE III.
EXCERPT OF DEVICE CHARACTERISTICS AND GROUPS

Group name	Characteristic name
Component Type	Input, Output
Device Type	Generic Device, Specialized Device
System Type	Application System, Operational System

TABLE II.
LISTING OF ENVIRONMENT GROUP AND CHARACTERISTICS

Group name	Characteristic name
Physical parameters	Temperature, Humidity, Ambient Noise, Visual comfort (Light luminance), Environment color

TABLE IV.
LISTING OF TASK GROUP AND CHARACTERISTICS

Group name	Characteristic name
Task parameters	Task, Goal, Sub-Task, Task Duration, Skill requirement, User, Resource, Environment

Other author, [25], presented a conceptualization for user interface development through ontological modeling, which provided descriptions of the user interface itself for a given architecture, for instance, in the description of components, relationships between components, among others. [26] presented a formal user interface representation describing both interaction and components aspects, while also discussed how the formal representation of user interface might be fit its development.

The authors [27] proposed a semantic framework in the context of the human system integration (HSI) research field. The main goal is extending the overall modeling capabilities in human-machine systems for improving human representation inside the overall system view. The work of [28] presented a particular implementation of user interface ontology tailored for the software development in avionics domain, where it might play a role as documentation resource.

III. THE DESIGN APPROACH

In this paper we propose cataloguing information on User, Device, Environment and Task domains for formalizing software models. This information includes the characteristics of the models, along with its descriptors. The goal of the taxonomy is assisting in answering the following questions:

- What human and device characteristics are important considering for user interface development?

- What are the literature references that support these human factors and device characteristics?

For assisting this task, the taxonomy presents a set of ten (10) descriptors, namely: name (a unique information that describes it); description (additional information for detailing the characteristic); unit of measurement (in which unit the characteristic is measured); the value space of the characteristic (*nominal, ordinal, interval, ratio, absolute*); taxonomy group (which category the characteristic belongs to); data type (what is the basic data type the characteristic presents - *string, enumeration, list/vector, real, integer*); how to measure/detect (how the characteristic information is measured or detected); reference/source (references/standards that supports the characteristic); and relations and comments (further explanations/observations from the given characteristic).

Table I presents excerpt for user characteristics. The complete listing is composed by a set of twenty-five (25) descriptors (for instance, related with physical body conditions), along with its seven (7) groups (encapsulating demographic, anthropometric, interaction related states, hearing, visual, motor and knowledge/experience data).

An excerpt (first level) composed by six (6) device descriptors, alongside its three (3) groups is presented (Table II). The complete work extends up to five other levels providing classification for components, devices and systems. Table III presents the Environment composed by one group and five (5) characteristics.

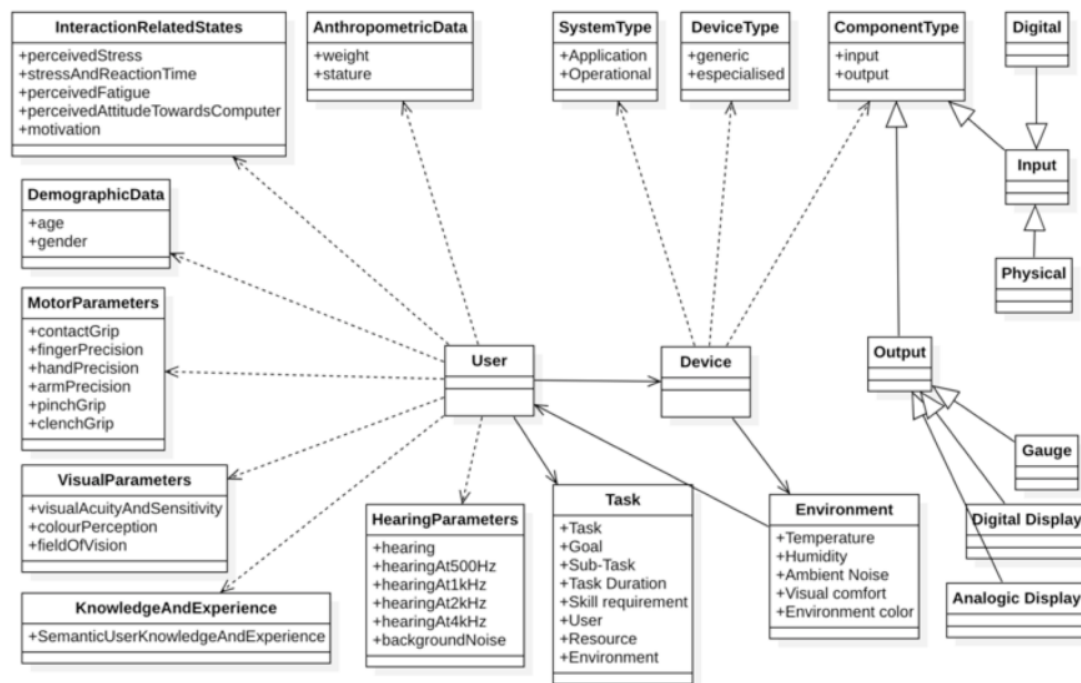


Fig 1. Domain model of user and device information system

The Task group and characteristics are listed in Table 1V, in which one group includes eight (8) characteristics. The next tasks in ontology creation are supported by the taxonomy, whose main role is ensuring required information for the ontology development phases:

- **Domain determination** – is performed by considering the need to know *User*, *Device*, *Environment* and *Tasks* aspects. The literature of human-machine interaction provides a number of measurements for estimating the quality of that interaction. In this context, one of the most widely accepted measurements considered is provided by the Ergonomics of human-system interaction's chapter of the International Organization for Standardization (ISO), which is standard ISO 9241-11. In this particular case the construct of *Usability* considers *User* and *Equipment* component, among others, as central for evaluating man-machine interaction aspects [29].
- **Terms enumeration** – important terms are enumerated for describing the domain or scope based on elements from the main taxonomy structure, i.e., taxonomy group names and taxonomy characteristics names.
- **Class and hierarchy definition** – a top-down approach might provide the first idea of a class and its sub-classes (i.e., hierarchies), given by examining the objects in the previously defined concepts from domain or scope enumeration.

- **Class properties definition** – definition of class properties (also referred to as slots) is given by examining the taxonomy-defined hierarchy.
- **Domain and range of properties** – the descriptors *Unit of measurement*, *Value space* and *Data type* of the taxonomy assist in defining facets.
- **Properties instances** – ultimately, individual instance(s) of class hierarchy and characteristics might be created.
- **Definition of relations** – based on the analysis of user and device models previously presented, the authors propose an extension of the relations framework provided by [30]. A list of 43 tailored relations along with their inversion relation names, are presented in the context of the interaction ontology.

The set of relation types presented, albeit useful for reasoning about the main characteristics of the proposed ontology, when depleting its capability of capturing the totality of properties and relations from existing models, can be extended and specialized for supporting additional and more representative semantics. Ontologies are on-going work, i.e., they require evolution with updated entities, properties and relations, and so are these relations classification. Additionally, we identified 171 mappings between relations and their source and destinations classes.

The goals of data properties are twofold. They are used for connecting classes to specific data values, supporting the parameters that compose the models, and restrict classes

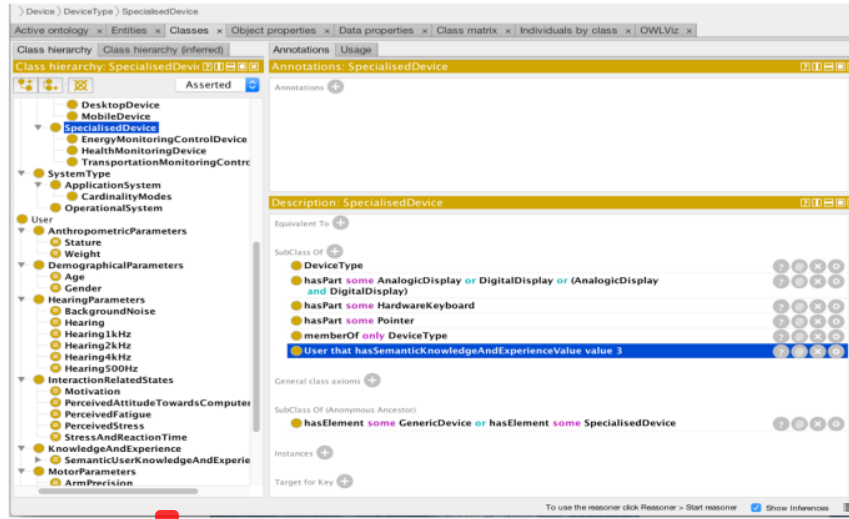


Fig 2. Representation of the expert requirement for specialized devices in Protégé Tool [21].

```

1  <!-- http://www.semanticweb.org/saulo/ontologies/2020/9/User-Ontology/Age -->
2
3  <owl:Class rdf:about="http://www.semanticweb.org/saulo/ontologies/2020/9/User-Ontology/Age">
4    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/saulo/ontologies/2020/9/User-Ontology#User"/>
5    <rdfs:subClassOf>
6      <owl:Restriction>
7        <owl:onProperty rdf:resource="http://www.semanticweb.org/saulo/ontologies/2020/9/User-Ontology/ageValue"/>
8        <owl:someValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
9      </owl:Restriction>
10   </rdfs:subClassOf>
11 </owl:Class>
12

```

Fig 3. Fragment of OWL code, for the User Device Ontology

named “Defined Classes” under necessary and sufficient conditions. We identified 64 data properties in the context of the current ontology. Next, a domain model is provided for representing the modeled objects (conceptual classes), as depicted in Figure 1.

Domain modeling organizes knowledge around the basic concepts that is investigated. The model is illustrated with a class diagram, which use classes and interfaces for capturing details about the entities that make up your system and the static relationships between them [31]. The reader will notice that the device portion is condensed for space constraints.

IV. RESULTS

The interaction ontology is developed in OWL language, based on RDF and RDF Scheme, and implemented in Protégé [21], presenting classes, properties and instances. The present ontology is heavy weighted (as opposite to those using controlled vocabularies), i.e., its internal structure represents semantic knowledge enriched with value and logical constraints. This particular type of ontology also has the potential of assisting designers in the requirements phase [32], [24]. Figure 2 depicts a fragment of the *Interaction Ontology*, highlighting the restrictions that ties all the ontol-

ogy parts together, i.e., the classes, sub-classes, class properties, range of properties, relations and data properties.

This particular instance highlights the *User* requirement for operating *Devices* of the type *Specialized Device*. The restriction demonstrates the requirement of a user with *Semantic Knowledge and Experience* rated as value 3, which according with [33] represents a domain expert user. The semantics of the restriction might also be expressed by the formal assertion in First Order Logic (FOL) notation presented in (1).

$$\forall (SpecialisedDevice \in Devices) \exists User \quad (1) \\ \exists hasSemanticKnowledgeAndExperienceValue(User) \\ \equiv 3$$

A snippet of the OWL code for the User Device Ontology is depicted in Figure 3, contemplating the definition of classes and sub-classes. A number of approaches exist for evaluating ontologies (e.g., development quality, mathematical correctness, metrics).

In this work, we make use of a *reasoner* for knowledge consistency checking, based on checking reflexive, transmission and redundancy properties of ontologies, providing contradictions implicit in the definitions, if any exist. In the case of the proposed ontology, logical consistency is evaluated against the HermiT OWL reasoned [34], and all detect-

ed classes are given as satisfiable. Additionally, descriptive metrics for the asserted classes of the ontology are presented in Table V.

TABLE V.
DESCRIPTIVE METRICS FOR THE ONTOLOGY'S ASSERTED CLASS HIERARCHY

Name	Metric
Axiom	1188
Logical axiom count	876
Declaration axioms count	312
Class count	76
Object property count	171
Data property count	64
Individual count	4
Annotation Property count	3
SubClassOf	172
EquivalentClasses	6

V. CONCLUSION AND FUTURE WORK

This work presented version of the interaction ontology, a structure tailored for contributing for the improvement of user interface design, considering semantic knowledge from human factors associated with users, tasks, and environment, along with knowledge from devices. By providing a heavy weighted ontology involving knowledge from the explored domains, the proposed work intends assisting designers in the constraints/requirements phase of user interface development, with potential of narrowing the gap between user interface design and implementation. Therefore, we achieved the goals of the work, namely, we discussed the necessity and usefulness of human factors and ergonomic knowledge during the user interfaces development; we described a multidimensional knowledge representation model related with human, device, task and environment models and we presented an ontology based on the models.

Future research includes improving the ontology, which is part of a research strand involving the design of an interactive systems evaluation platform, and submitting it for a journal. We also intent extending the validation method for the ontology, i.e., converting the protégé ontology for Alloy notation, which will allow testing instances and validate formal assertions about the represented knowledge. Counter-examples might be useful on the validation of mathematical/logical assertions. Additionally, moving to a realistic case study will allow the authors inform and validate the use of such ontologies in the development of user interfaces for interactive systems.

ACKNOWLEDGMENTS

The authors would like to acknowledge Conselho Nacional de Pesquisa (CNPq), COMPETE: POCI-01-0145-FEDER-007043, and FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020. We also acknowledge the support of Instituto Federal de Educação, Ciência e Tecnologia de Goiás (IFG).

REFERENCES

- [1] Benyon, D. (2014). Designing interactive systems: A comprehensive guide to HCI, UX and interaction design.
- [2] Leveson, N. (2011). Engineering a safer world: Systems thinking tied to safety. MIT press.
- [3] Gruber, T. (1993). A translation approach to portable ontologies. Knowledge Acquisition, 19(1), 220.
- [4] Hesse, W. (2005, June). Ontologies in the Software Engineering Process. In EAI (pp. 3-16).
- [5] Tanasescu, V. (2004). An ontology-driven life-event portal. Master. Computer Science.
- [6] Kaklanis, N., Biswas, P., Mohamad, Y., Gonzalez, M. F., Peissner, M., Langdon, P., ... & Jung, C. (2016). Towards standardisation of user models for simulation and adaptation purposes. Universal Access in Information Society, 15(1), 21-48.
- [7] Eason, K. D. (1991). Ergonomic perspectives on advances in human-computer interaction. Ergonomics, 34(6), 721-741.
- [8] Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2003). Human-computer interaction. Pearson Education.
- [9] Hathaway, W. E. (1988). Educational facilities: Neutral with respect to learning and human performance?. CEFJJournal, 26(4):8-12.
- [10] Wang, H. and Russ, R. R. (2008). Computer classroom wall colour preference and the relationship with personality type of college students. Colour: DesignCreat, 2(4):1-13.
- [11] Proctor, R. W., Vu, K.-P. L., and Seel, N. (2012). Human information processing. Encyclopedia of the Sciences of Learning, pages 1458-1469.
- [12] Bickhard, M. H. (1992). How does the environment affect the person. Children's development with in social context, 1:63-92.
- [13] Storrs, G. (1995). The notion of task in human-computer interaction. IJCSHCI, pages 357-365.
- [14] Bergh, J., Sahni, D., & Coninx, K. (2010). Task Models for Safe Software Evolution and Adaptation. In Task Models and Diagrams for User Interface Design.
- [15] Happel, H. J., & Seedorf, S. (2006, November). Applications of ontologies in software engineering. In Proc. of Workshop on Semantic Enabled Software Engineering (SWESE) on the ISWC (pp. 5-9).
- [16] Milligan, C., Halladay, S. (2003). "The Realities and Facilities Related to Knowledge Representation," IPSI Belgrade, Proceedings of the IPSI-2003 Montenegro conference, Sveti Stefan, Montenegro.
- [17] Chan, C. W. (2004, May). The knowledge modeling system and its application. In Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No. 04CH37513) (Vol. 3, pp. 1353-1356). IEEE.
- [18] Gašević, D., Djuric, D., & Devedžić, V. (2009). Model driven engineering and ontology development. Springer Science & Business Media.
- [19] Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies: where is their meeting point?. Data & Knowledge Engineering, 46(1), 41-64.
- [20] Staab, S., & Studer, R. (Eds.). (2010). Handbook on ontologies. Springer Science & Business Media.
- [21] Noy, N. F., & McGuinness, D. L. (2004). Ontology development 101: A guide to creating your first ontology. 2001. See <http://protege.stanford.edu/publications>.
- [22] Falbo, A. (1998). Integração de Conhecimento em um Ambiente de Engenharia de Software (Doctoral dissertation, Tese de Doutorado em Engenharia de Software, Universidade Federal do Rio de Janeiro, Rio de Janeiro).
- [23] Benjamin, P. C., Menzel, C. P., Mayer, R. J., Fillion, F., Futrell, M. T., de Witte, P. S., & Lingineni, M. (1994). Idef5 method report. Knowledge Based Systems, Inc.
- [24] Ruiz, F., & Hilera, J. R. (2006). Using ontologies in software engineering and technology. In Ontologies for software engineering and technology (pp. 49-102). Springer, Berlin, Heidelberg.
- [25] Shahzad, S. K. (2011). Ontology-based User Interface Development: User Experience Elements Pattern. J. UCS, 17(7), 1078-1088.
- [26] Paulheim, H., & Probst, F. (2011). A Formal Ontology on User Interfaces-Yet Another User Interface Description Language?: Position Paper. In CEUR Workshop Proceedings (Vol. 747, pp. Paper-9). VTH.
- [27] Orellana, D. W., & Madni, A. M. (2014). Human system integration ontology: enhancing model based systems engineering to evaluate human-system performance. Procedia Computer Science, 28, 19-25.

- 2
- [28] Tan, H., Adlemo, A., Tarasov, V., & Johansson, M. E. (2017). Evaluation of an application ontology. In Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology Bologna-Bolzano, Italy, September 21–23, 2017 (Vol. 2050). CEUR-Ws.
- [29] ISO/IEC (International Organization for Standardization) (1998). Standard 9241: Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s, Part 11. Guidance on Usability. Available at: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en> (accessed in 20 Jan 2019).
- [30] Štorga, M., Marjanović, D., & Andreasen, M. M. (2007, January). Relationships between the concepts in the design ontology. In 16th International Conference on Engineering Design–ICED 07.
- [31] Pilone, D., & Pitman, N. (2005). UML 2.0 in a Nutshell. " O'Reilly Media, Inc. ".
- [32] Decker, B., Ras, E., Rech, J., Klein, B., & Hoecht, C. (2005, November). Self-organized reuse of software engineering knowledge supported by semantic wikis. In Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE) (p. 76). ISWC Galway, Ireland.
- [33] Karwowski, W. (2006). International Encyclopedia of Ergonomics and Human Factors, -3 Volume Set. CRC Press.
- [34] Glimm, B., Horrocks, I., Moeller, S., Stoilos, G., & Wang, Z. (2014). HermiT: an OWL 2 reasoner. Journal of Automated Reasoning, 53(3), 245-269.

74%

SIMILARITY INDEX

PRIMARY SOURCES

1	Saulo Silva, Orlando Belo. "Chapter 36 Domain and Semantic Modeling in the Context of Interactive Systems Development: User and Device Cases", Springer Science and Business Media LLC, 2023 Crossref	3441 words — 67%
2	www.emerald.com Internet	38 words — 1%
3	scholarworks.waldenu.edu Internet	33 words — 1%
4	etd.adm.unipi.it Internet	30 words — 1%
5	es.scribd.com Internet	27 words — 1%
6	journals.sagepub.com Internet	27 words — 1%
7	www.igi-global.com Internet	26 words — 1%
8	www.acarindex.com Internet	23 words — < 1%
9	www.ksi.edu Internet	

22 words — < 1%

10 pt.scribd.com
Internet

18 words — < 1%

11 www.tandfonline.com
Internet

17 words — < 1%

12 libeprints.open.ac.uk
Internet

16 words — < 1%

13 Vasco Santos, Orlando Belo. "Modelling ETL Conciliation Tasks Using Relational Algebra Operators", 2014 European Modelling Symposium, 2014
Crossref

11 words — < 1%

14 downloadmedicalbooks.files.wordpress.com
Internet

11 words — < 1%

15 www.aic-color.org
Internet

10 words — < 1%

16 ebin.pub
Internet

9 words — < 1%

17 repositorio.unesp.br
Internet

9 words — < 1%

18 dokumente.ub.tu-clausthal.de
Internet

8 words — < 1%

19 espace.curtin.edu.au
Internet

8 words — < 1%

20 uwe-repository.worktribe.com
Internet

8 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	OFF	EXCLUDE MATCHES	OFF