Simone Dornelas Costa

An Ontology Network to support Knowledge Representation and Semantic Interoperability in the HCI Domain

Vitória, ES

2022

Simone Dornelas Costa

An Ontology Network to support Knowledge Representation and Semantic Interoperability in the HCI Domain

Tese de Doutorado submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Doutor em Ciência da Computação.

Universidade Federal do Espírito Santo

Centro Tecnológico

Programa de Pós-Graduação em Informática

Supervisor: Monalessa Perini Barcellos

Co-supervisor: Ricardo de Almeida Falbo (*in memoriam*)

Vitória, ES

2022

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

C837a

Costa, Simone Dornelas, 1987-
An Ontology Network to support Knowledge Representation and Semantic Interoperability in the HCI Domain / Simone Dornelas Costa. - 2022.
205 f. : il.

Orientadora: Monalessa Perini Barcellos.
Coorientador: Ricardo de Almeida Falbo (in memoriam).
Tese (Doutorado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Human-Computer Interaction. 2. Ontology. 3. Ontology Network. I. Barcellos, Monalessa Perini. II. Falbo (in memoriam), Ricardo de Almeida. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

An Ontology Network to support Knowledge Representation and Semantic Interoperability in the HCI Domain

Simone Dornelas Costa

Tese submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Aprovada em 08 de julho de 2022.

Profa. Dra. Monalessa Perini Barcellos (UFES)
Orientadora, participação remota

Prof. Dr. Vítor E. Silva Souza (UFES)
Membro Interno, participação remota

Prof. Dr. João Paulo A. Almeida (UFES)
Membro Interno, participação remota

Profa. Dra. Luciana Aparecida Martinez Zaina (UFSCar)
Membro Externo, participação remota

Prof. Dr. Roberto Pereira (UFPR)
Membro Externo, participação remota

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Vitória/ES, 08 de julho de 2022

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
MONALESSA PERINI BARCELLOS - SIAPE 1490891
Departamento de Informática - DI/CT
Em 11/07/2022 às 12:17

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/512130?tipoArquivo=O

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
VITOR ESTEVAO SILVA SOUZA - SIAPE 2525114
Departamento de Informática - DI/CT
Em 11/07/2022 às 14:39

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/512366?tipoArquivo=O

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
JOAO PAULO ANDRADE ALMEIDA - SIAPE 1516048
Departamento de Informática - DI/CT
Em 11/07/2022 às 16:16

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/512517?tipoArquivo=O

Dedication

To my beloved grandparents, Florentino and Irma.
To my beloved parents, José and Jane.
To my beloved brother, Leonardo.

Acknowledgements

At this juncture, I have many people to thank with open arms for their help, and probably words will fall short.

It is a genuine pleasure to express my deep sense of thanks and gratitude to my co-supervisor, Prof. Dr. Ricardo de Almeida Falbo (*in memoriam*). His dedication and keen interest above all his overwhelming attitude to help his students had been mainly responsible for completing my work.

I owe a deep sense of gratitude to my supervisor Prof. Dr. Monalessa Perini Barcellos for the continuous support of my Ph.D. study and research. Her patience, prompt inspiration, immense knowledge, timely suggestions with kindness, enthusiasm, dynamism, and plentiful experience have encouraged me all the time in my academic research and daily life.

Falbo and Mona, I will always be grateful for having the opportunity to study under you. I could not have imagined having better advisors for my Ph.D. study. It is not sufficient to express my gratitude with only a few words.

It is a great pleasure to express my special thanks to the committee members for their insightful comments, hard questions, and making critical suggestions to improve this research.

I would like to extend my sincere thanks to NEMO colleagues for their insightful comments, discussions, suggestions, friendship, and constant encouragement throughout my research period. With special thanks to Fabiano, Murilo, and Paulo who were a constant source of reference. I wish to extend my sincere thanks to Carolina for having carried out the development of UXON with excellence.

I am deeply grateful to Prof. Dr. Tayana Conte, Prof. Dr. Kathia Oliveira, and Prof. Dr. Luciana Zaina for their technical and expert support and valuable contribution to the evaluation of the ontologies.

I am deeply grateful to NEMO professors for their assistance, intellectual support, and kind help at every stage of this research.

I would like to offer my special thanks to USES colleagues: Leonardo, Mauro, Prof. Dr. Tayana Conte, and Prof. Dr. Bruno Gadelha for their partnership and contribution to this research.

My sincere thanks also go to my UFES colleagues for allowing me to work partially exclusively on this research.

Finally, I would like to express my gratitude to my parents, my partner and friends (specially Marilia and Luiz) for believing in me. Without their love, tremendous understanding, and encouragement over the past few years, it would be impossible for me to complete my study.

Resumo

A Interação Humano-Computador (IHC) é uma área de conhecimento multidisciplinar voltada para a prática da tecnologia da informação centrada no ser humano. Atualmente, cada vez mais os sistemas interativos precisam ser personalizados, responsivos, adaptáveis, fáceis de usar e caracterizados por ambientes cada vez mais conectados e aplicativos inteligentes. Sendo uma área tão diversa, a IHC envolve um conjunto diversificado e complexo de conhecimentos e conceitos. No entanto, isso leva a problemas de interoperabilidade semântica, que afetam usuários, designers e sistemas interativos. Além disso, o projeto e a avaliação de sistemas interativos são atividades intensas em termos de conhecimento. Nesse contexto, problemas relacionados ao conhecimento, como os referentes à sua representação e compartilhamento, acarretam o risco de perda de conhecimento relevante. Portanto, são necessários mecanismos eficientes para promover o entendimento comum e a construção coletiva do conhecimento. Ontologias têm sido utilizadas com sucesso em diversos domínios para capturar e organizar o conhecimento, buscando lidar com a interoperabilidade e problemas relacionados ao conhecimento. Portanto, neste trabalho, argumentamos que organizar ontologias de IHC em uma rede de ontologias: fornece uma conceituação abrangente do domínio IHC; favorece o crescimento, a reutilização e a integração do conhecimento; potencializa o uso de ontologias em soluções baseadas em conhecimento e interoperabilidade. Nesse sentido, propusemos a Rede de Ontologias de Interação Humano-Computador (HCI-ON). A HCI-ON é baseada em uma ontologia de fundamentação e aborda importantes aspectos de IHC alinhados com seus padrões e literatura. A HCI-ON possui uma ontologia central que aborda o fenômeno de interação humano-computador e ontologias específicas de domínio que abrangem subdomínios de IHC, como design, avaliação e interface com o usuário, entre outros. Visando o crescimento consistente do conhecimento, a HCI-ON disponibiliza mecanismos para suportar sua constante evolução através da adição de ontologias novas ou existentes. No domínio de IHC é possível utilizar a HCI-ON como um todo ou em extratos, para resolver problemas de interoperabilidade semântica e relacionados ao conhecimento. Para demonstrar o uso da HCI-ON para apoia a resolução de tais problemas, usamos um extrato HCI-ON para desenvolver UXON, um sistema que apoia a avaliação de UX com base em dados no registro de interações produzidos por um aplicativo imersivo chamado Compomus. Como resultado, o uso do HCI-ON foi considerado viável e útil.

Palavras-chaves: Interação Humano-Computador. Ontologia. Rede de Ontologia.

Abstract

Human-Computer Interaction (HCI) is a multidisciplinary knowledge area aimed at the practice of information technology centered on humans. Currently, increasingly interactive systems are required to be personalized, responsive, adaptive, user-friendly, and characterized by increasingly connected environments and intelligent applications. Being such a diverse area, HCI involves a diverse body of knowledge and a complex set of concepts. This leads to semantic interoperability problems, which affect users, designers, and interactive systems. Moreover, the design and evaluation of interactive systems are knowledge-intensive activities. In this context, knowledge-related problems, such as the ones related to knowledge representation and sharing, cause the risk of losing relevant knowledge. Therefore, efficient mechanisms to promote common understanding and collective construction of knowledge are necessary. Ontologies have been successfully used in several domains to capture and organize knowledge seeking to deal with interoperability and knowledge-related problems. In this work, we argue that organizing HCI ontologies in an ontology network provides a comprehensive conceptualization of the HCI domain; favors knowledge growth, reuse, and integration; and potentializes the use of ontologies in knowledge-based and interoperability solutions. In this sense, this work proposes the Human-Computer Interaction Ontology Network (HCI-ON). HCI-ON is grounded on a foundational ontology, is aligned with HCI standards and literature, and addresses HCI relevant aspects. HCI-ON has a core ontology that addresses the human-computer interaction phenomenon and domain-specific ontologies covering HCI subdomains such as HCI design, evaluation, and user interface, among others. Aiming at knowledge growth in a consistent way, HCI-ON provides mechanisms to support its constant evolution throughout the addition of new or existing ontologies. In the HCI domain is possible to use HCI-ON as a whole or extracts of it to solve semantic interoperability and knowledge-related problems. To demonstrate the use of HCI-ON to support solving such problems, we used an HCI-ON extract to develop UXON, a system that supports UX evaluation based on interaction logging produced by an immersive application called Compomus. As a result, the use of HCI-ON was considered feasible and useful.

Keywords: Human-Computer Interaction. Ontology. Ontology Network.

List of Figures

List of Tables

List of abbreviations and acronyms

ADO	Aligned Domain-specific Ontology
AUI	Adaptive User Interface
COM	Core Ontology on Measurement
CQ	Competency Question
CSDO	Computer System Design Ontology
CUO	Context of User Ontology
DSR	Design Science Research
ETL	Extract, Transform, and Load
GUI	Graphical User Interface
HCD	Human Centred Design
HCI	Human-Computer Interaction
HCI-ON	Human-Computer Interaction Ontology Network
HCIDO	Human-Computer Interaction Design Ontology
HCIEO	Human-Computer Interaction Evaluation Ontology
HCIMO	HCI Modality Ontology
HCIO	Human-Computer Interaction Ontology
HCIQCO	HCI Quality Characteristics Ontology
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
KM	Knowledge Management
KTID	Knowledge Tool for Interaction Design
MVP	Minimum Viable Product
OBA	Ontology-Based Architectures

ODD	Ontology-Driven Development
OEA	Ontology-Enabled Architectures
OED	Ontology-Enabled Development
ON	Ontology Network
OWL	Web ontology Language
RDF	Resource Description Framework
RQ	Research Question
RSRO	Reference Software Requirements Ontology
SEON	Software Engineering Ontology Network
SEVOCAB	Software and System Vocabulary
SLR	Systematic Literature Review
SM	Systematic Mapping
SNOPI	Social Network with Ontology-based adaPtive Interface
SO	Specific Objective
SPARQL	SPARQL Query Language for RDF
SPO	Software Process Ontology
SysSwO	System and Software Ontology
Turtle	Terse RDF Triple Language
UCD	User Centered Design
UCD	User Centred Design
UCO	User Characterization Ontology
UFO	Unified Foundational Ontology
UI	User Interface
UIT&EO	UI Types and Elements Ontology
USES	Usability and Software Engineering Research Group
UX	User eXperience

UXON	User eXperience Ontology Network-based system
W3C	World Wide Web Consortium
WDO	Well-founded Domain-specific Ontology
WIMP-UI	Window Icon Menu Pointer User Interface

Contents

1 Introduction

This chapter presents an overview of this thesis and defines the basis for the following chapters. It discusses the research context and motivation, research hypothesis, objectives and methodological aspects that have guided the work. Last, it presents the structure of this document.

1.1 Context and Motivation

Human-Computer Interaction (HCI) is a multidisciplinary knowledge area that aggregates a vast and multifaceted community, bound by the evolving concept of usability, and the integrating commitment to value human activity and experience as the primary driver in technology (Carroll, 2014). It presents as a major challenge to the practice of information technology centered on the human being and his/her values (Harper et al., 2008). It is connected to other research areas, involving knowledge from multiple fields, such as ergonomics, cognitive science, user experience, human factors, among others (Sutcliffe, 2014).

In the last years, the HCI area has expanded to include a myriad of interaction techniques and devices, becoming a host for emerging ubiquitous, smart environments, handheld and context-aware interactions. Being such a diverse area, it involves a diverse body of knowledge and the landscape of HCI concepts is complex. In addition, as the HCI area continues to mature, new terms are proposed and new meanings are assigned to existing terms. As a consequence, it is not trivial to have a common conceptualization of HCI, leading to semantic interoperability problems (such as ambiguity and imprecision when interpreting shared information) and hampering communication and knowledge transfer (Carroll, 2014; Preece; Sharp; Rogers, 2015).

Interoperability refers to the ability to interact, in order to achieve some objective. It is a comprehensive problem that, in general, involves two or more artifacts (systems, components, models, documents, standards, databases, etc.), among which some form of interaction is intended (using functionalities, providing and using services, transferring data, exchanging information and knowledge, etc.) to achieve a goal, requiring little or no knowledge about the characteristics of the artifacts involved in the integration (Pokraev, 2009).

The growth and expansion of the HCI area have caused the emergence of many references (books, standards, research papers, etc.) coming from different sub-communities, not sharing a common conceptualization. The lack of a common conceptualization causes misunderstanding and interoperability problems when dealing with those references. Rusu et al. (2015) argues that "the lack of generally agreed formal definitions of HCI/usability/UX may have consequences on their development and recognition among Computer Science communities, especially in regions where HCI is poorly developed". Many times, even references from the same sub-community are not harmonized. For example, the Software and Systems Engineering Vocabulary (SEVOCAB) (ISO/IEC/IEEE, 2017) presents three slightly different definitions for "user interface": (i) all components of an interactive system (software or hardware) that provide information and controls for the user to accomplish specific tasks with the interactive system (ISO/IEC, 2014); (ii) the ensemble of software and hardware that allows a user to interact with a system (ISO/IEC, 2008b; ISO/IEC/IEEE, 2018); (iii) the interface that enables information to be passed between a human user and hardware or software components of a computer system (ISO/IEC, 2012). Definition (i) explicitly mentions interactive system (e.g., tablets, smart phones, wearable devices), while definitions (ii) and (iii) refer respectively to system and computer system (e.g., servers). In these definitions, do interactive system, computer system and system have the same meaning? This is not clear. As a consequence, it is not clear if user interface exists only in interactive systems or in any kind of computer system or system. It is worth pointing out that the three definitions come from different standards defined by the same sub-committee (SC 7 - Software and Systems Engineering) of the Joint Technical Committee JTC 1 (Information Technology) that joins two standardization organizations, namely the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

Different standards often adopt different terminologies and, sometimes, it is necessary to use different standards in a combined way. That means, it is necessary to harmonize different terminologies/conceptualizations so that different standards can work together and knowledge workers can better understand each other. As another example of semantic conflict, consider the following definitions provided by different standards and referring, respectively, to "dialog" and "user interaction": (i) interaction between a user and an interactive system as a consequence of user actions (inputs) and system responses (output) in order to achieve a goal (ISO/IEC, 2016) and (ii) exchange of information between a user and an interactive system via the user interface to complete the intended task (ISO, 2019b). Although the different standards use different terms to these definitions, they seem to refer to the same concept.

With the emergence of the Internet of Things and technologies associated with smart environments and user-adaptive systems, we are moving towards smart interactive computer systems characterized by full integration and semantics (Pourzolfaghar; Helfert, 2016).

Semantic interoperability problems in the HCI area are frequent and affect users, designers and interactive systems. For instance, nowadays, people use and need to cope with lots of different user interfaces during the workday. It leads to socio-cognitive dimensions of interoperability, such as information sharedness, operational awareness, communication readiness, adaptiveness and coupledness (Clemmensen, 2018). Moreover, most user interfaces are not built in isolation, but sit atop a collection of software libraries, toolkits, protocols, and standards typically inaccessible to user-centred design processes (Edwards; Newman; Poole, 2010). In addition, different systems represent the same data in different ways, using different syntactic and conceptual structures, terminologies or different interpretations of the same terminology

(Staab; Stuckenschmidt, 2006). To solve interoperability problems like these, agreement and clarification of the data are necessary.

Even more interactive systems are required to be personalized, responsive, adaptive, simple and characterized by increasingly connected environments and intelligent applications. Moreover, to deal with structural, syntactic and semantic interoperability in order to function as desired by the user or the group of user becomes almost inherent in these systems (Carmagnola; Cena; Gena, 2011). Although technologies and standards for interoperability have been built, a challenge still remains for the design, integration and interoperability of the various elements that make up this kind of interactive systems (Pourzolfaghar; Helfert, 2016).

Interoperability issues are also perceived in the context of user-adaptive systems. Due to the lack of interoperability and synchronization between these systems, each system needs to build its user model separately. Therefore, there is a strong need for the next generation of user models to be interoperable, i.e., to be able to exchange portions of the user model and use the information that has been exchanged to enrich user experience (Carmagnola; Cena; Gena, 2011).

In HCI, product design and evaluation involve highly specialized knowledge (e.g., human factors, ergonomics and user-system interaction design) and can be considered knowledge-intensive activities. In addition, HCI and human-centred design of interactive systems comprises a large amount and diversity of information and knowledge about the interplay among the users, tasks, task contexts, user's interaction, information technology, and the environments in which the systems are used (Bigham; Bernstein; Adar, 2015; Kendall; Kendall, 2008; Pourzolfaghar; Helfert, 2016). Thus, to be effective, projects require efficient mechanisms for the collective construction of the knowledge of those involved and to support the collective understanding regarding the purpose of the software and its context of use.

In the human-centred design process, tacit knowledge is dominant (Ibrahim; Fay, 2006). Domain knowledge or knowledge about systems and artifacts are exchanged among many stakeholders (Pourzolfaghar; Helfert, 2016). However, this knowledge transfer is characterized with high risk of losing tacit knowledge intrinsic to human-human interactions. Knowledge Management (KM) seeks to transform tacit and individual knowledge into explicit and shared knowledge. In order to raise individual knowledge to an organizational level, KM aims to promote the dissemination of knowledge and learning, making knowledge accessible and reusable throughout the organization (O'Leary, 1998; Rus; Lindvall, 2002; Schneider, 2009).

For KM, as important as knowledge itself is its formalization, i.e. the form of its representation and storage must be in such a way that promotes knowledge capture and sharing (Schneider, 2009). For that, knowledge should be represented using a shared conceptualization (Castro et al., 2020).

Since HCI is a complex knowledge area, ontologies can be useful to capture, organize and share the involved knowledge. An ontology is a formal and explicit specification of a shared conceptualization (Gruber, 1995). Ontologies have been successfully used in several domains (e.g., IT Service Management (Pardo et al., 2013), E-commerce (Tamma et al., 2005), Health (Liyanage; Krause; Lusignan, 2015; Sene; Kamsu-Foguem; Rumeau, 2018), Oil and Gas (Carbonera; Abel; Scherer, 2015) and Education (Yago et al., 2018)) to capture and organize knowledge aiming to deal with interoperability and knowledge-related problems, as the ones aforementioned.

In the HCI domain, ontologies have been used mainly in knowledge management and reasoning solutions addressing user interface, user interaction and user characterization (Costa; Barcellos; Falbo, 2021). HCI ontologies have covered different HCI aspects, sometimes overlapping concepts and scope. Moreover, there are inconsistencies among their conceptualizations, even in HCI core concepts. This suggests lack of a common understanding of the HCI phenomenon. Furthermore, HCI ontologies have been developed to solve specific problems for the purpose of practical applications, in specific and isolated contexts. Hence, they represent a biased view in its practical application instead of a reference view of the addressed portion of the HCI domain (Costa et al., 2020; Costa; Barcellos; Falbo, 2021). This leads to problems related to knowledge integration.

Ideally, in wide domains like HCI, ontologies should not be stand-alone artifacts. Representing the domain knowledge as a single ontology results in a large and monolithic ontology that is hard to build, manipulate, use and maintain (Suárez-Figueroa et al., 2012). Hence, they should relate to each other, forming a network of interlinked semantic resources, i.e., an Ontology Network (ON). ONs enable to establish a comprehensive conceptualization that provides a common understanding of the domain of interest and can be used as a reference to solve semantic interoperability and knowledge problems related to the conceptualization as a whole or to extracts of it. Hence, integrating several ontologies into an ON provides a framework that can be explored to potentialize and increase the set of solutions in the universe of discourse addressed by the ON (Ruy et al., 2016).

In view of the above, this research focuses on the problem of interoperability and knowledge representation in the Human-Computer Interaction domain, precisely on establishing a domain conceptualization in such a way that promotes knowledge sharing, reuse and integration. We believe that organizing HCI ontologies as an ontology network provides a comprehensive conceptualization of the HCI domain and that the ontology network can figure as a reference knowledge framework.

1.2 Research Hypothesis

Considering, as previously discussed, that:

- HCI is a multidisciplinary area that involves highly specialized knowledge;

- There are inconsistencies among HCI standards conceptualizations, even in HCI core concepts;
- The absence of a consensual and shared conceptualization can hamper communication and knowledge transfer;
- It is not trivial to have a common conceptualization of HCI because it is continuously evolving, which leads to new terms and new meanings assigned to the existing terms;
- HCI is a knowledge-intensive process in which user interfaces are not built in isolation, requiring interoperability among the used resources;
- HCI encompasses a myriad of interaction techniques and devices, leading to semantic interoperability problems;
- Interactive system users use many different user interfaces and need to experience interoperability in the context of information sharedness, operational awareness, communication readiness, adaptiveness, and coupledness;
- Ontologies are suitable instruments for addressing semantic issues;
- Ontology networks allow organizing ontologies, providing comprehensive, consistent and integrated knowledge;

the research hypothesis of this thesis is:

Knowledge-related and semantic interoperability problems in the HCI domain can be addressed with a consensual, shared and comprehensive conceptualization of HCI, represented by means of an ontology network.

1.3 Objectives

The general objective of this thesis is to *propose a knowledge framework of the HCI domain, which provides a comprehensive conceptualization of that domain, favors knowledge growth, reuse and integration, and supports knowledge-based and interoperability solutions*. This general objective is broken down into the following specific objectives (SO):

- **SO1. Investigate the state of the art about ontologies in the HCI domain**: this goal concerns with identifying ontologies that address HCI aspects and finding out how they have been developed and used to solve problems in the HCI domain.
- **SO2. Establish a HCI reference knowledge framework**: it aims to develop an ontology network to establish a consensual, shared and comprehensive conceptualization of the HCI domain that can be used to support knowledge-related and interoperability problems and

covers central aspects of the HCI domain, such as the HCI phenomenon, human-centred processes (e.g., design and evaluation), HCI techniques (e.g., Persona) and other aspects relevant to the domain, such as user interface, usage context, user characterization, among others.

- **SO3. Establish mechanisms to support knowledge access, creation, integration and evolution**: it concerns the reuse and growth of knowledge available in the framework, by means of the creation of new knowledge or integration of existing knowledge.

- **SO4. Apply the knowledge framework to solve knowledge-related and interoperability problems in the HCI domain**: concerns the use of the knowledge framework (as a whole or extracts of it) to solve semantic interoperability and knowledge-related problems in the HCI domain.

1.4 Research Method

The research method adopted in this work follows the Design Science Research (DSR) paradigm, which concerns extending human and organizational capabilities by creating new and innovative artifacts (Hevner et al., 2004; Hevner, 2007). DSR is an iterative process involving three cycles (Hevner, 2007): *Relevance Cycle*, *Design Cycle* and *Rigor Cycle*.

Relevance is mainly related to: (i) research motivation, which arises from needs or possible improvement opportunities in current theories, as well as (ii) good articulation between the proposed solution and the motivation, to reinforce the contributions. *Rigor* is associated with the use of a reliable body of knowledge (e.g., theories, methods, models, experiences, and expertise) in the research effort. Finally, *Design* concerns the core activities of the research process towards achieving the research objectives and supporting the research hypothesis. As such, Design takes Relevance and Rigor aspects into account. Figure 1.1 summarizes the Design Science cycles performed in this research.

A Design Science Research project begins with the ***Relevance Cycle***, which involves defining the application context, problem to be addressed, the research requirements and the criteria for evaluating the results (Hevner, 2007). The **motivation** for this work was summarized in the topics pointed out in Section 1.1. The **problem** addressed by this work regards knowledge sharing and semantic interoperability in HCI, which involves the need for a comprehensive and consistent conceptualization of the HCI domain, to be used as a knowledge framework for communication, learning and interoperability purposes. The problem was identified by analyzing the literature (we discuss this further in Chapter 3) and from practical experiences of the authors when dealing with different HCI standards and other knowledge sources. For example, when working with HCI and Software Engineering professionals, we noticed that people with different backgrounds (e.g., Industrial Design, Web Design, Software Engineering) had different understandings of the same HCI concept. For instance, to talk about the design of an interactive

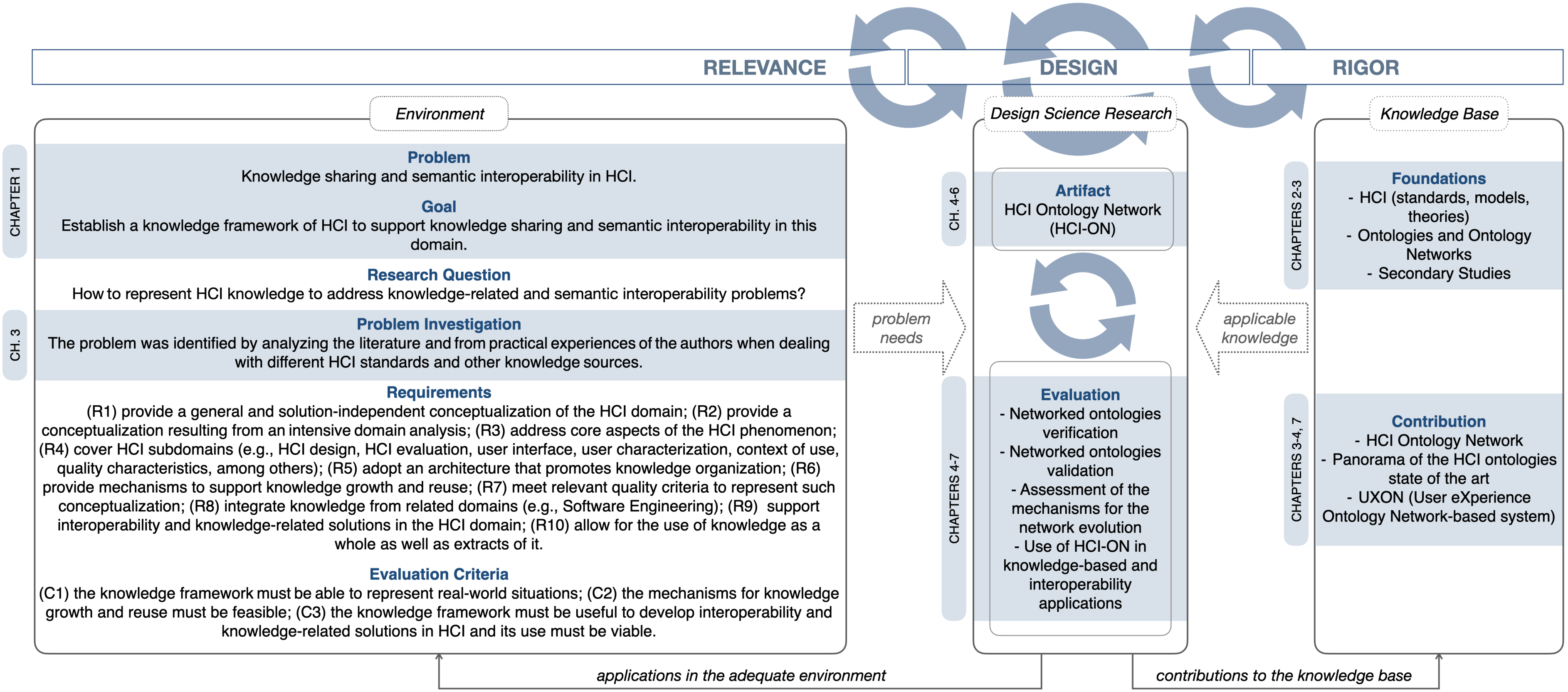


Figure 1.1 – Overview of the Design Science cycles in this research (based on (Hevner et al., 2004; Hevner, 2007)).

computer system, it is necessary to understand (and agree on) what an interactive computer system is and how and why user and system interact, as well as what an HCI requirement is and how it can be met by a design solution. This way it is possible to properly design HCI solutions to meet the user requirements.

The **research question** posed according to the aforementioned problem is: How to represent HCI knowledge to address knowledge-related and semantic interoperability problems? Therefore, our **goal** was to establish a knowledge framework of HCI to support knowledge sharing and semantic interoperability in this domain. As **requirements**, we established that the framework must: (R1) provide a general and solution-independent conceptualization of the HCI domain; (R2) provide a conceptualization resulting from an intensive domain analysis; (R3) address core aspects of the HCI phenomenon; (R4) cover HCI sub-domains (e.g., human-centred design and evaluation, user interface, user characterization, context of use, quality characteristics, among others); (R5) adopt an architecture that promotes knowledge organization; (R6) provide mechanisms to support knowledge growth and reuse; (R7) meet relevant quality criteria to represent such conceptualization; (R8) integrate knowledge from related domains (e.g., Software Engineering); (R9) support interoperability and knowledge-related solutions in the HCI domain; (R10) allow for the use of knowledge as a whole as well as extracts of it. In addition to the requirements to be met by the produced artifact, we defined the following evaluation criteria: (C1) the knowledge framework must be able to represent real-world situations; (C2) the mechanisms for knowledge growth and reuse must be feasible; (C3) the knowledge framework must be useful to develop interoperability and knowledge-related solutions in HCI and its use must be viable.

The **Design Cycle** involves developing and evaluating artifacts or theories to solve the identified problem (Hevner, 2007). Therefore, in this cycle, we developed and evaluated the *HCI Ontology Network* (HCI-ON). We decided to use an ontology network to implement the knowledge framework because, as we discussed in Section 1.1, ontologies are suitable instruments for addressing semantic issues and ontology networks allow organizing them in such a way that results in comprehensive, consistent and integrated knowledge. Our motivation to develop an ontology network was reinforced by the results of a systematic investigation we performed (presented in detail in Chapter 3), which showed us inconsistencies among the conceptualization of existing HCI ontologies and a focus on developing ontologies biased in the applications they support (Costa; Barcellos; Falbo, 2021). This hampers ontology reuse and integration to support solutions to more comprehensive HCI problems.

To meet R1, the HCI-ON ontologies are reference ontologies, i.e., a special kind of conceptual model representing a model of consensus within a community. It is a solution-independent specification with the aim of making a clear and precise description of domain entities for the purposes of communication, learning and problem-solving (Guizzardi, 2007). In addition, aiming to define a reliable conceptualization, as defended by Guarino (1998), the Unified Foundational Ontology (UFO) (Guizzardi, 2005; Guizzardi; Falbo; Guizzardi, 2008;

Guizzardi et al., 2013; Guizzardi et al., 2015; Guizzardi et al., 2018) was used to provide the ontological foundations for the network. Regarding R2, the developed ontologies are based on HCI literature (presented in Chapter 2) and HCI domain experts. To satisfy R3, we developed the Human-Computer Interaction core Ontology (HCIO) (presented in Chapter 5), which addresses core aspects of the HCI phenomenon that are shared with the other HCI-ON ontologies (Costa et al., 2022). Concerning R4, we developed domain ontologies addressing HCI design and evaluation (presented in Chapter 6), and have worked on the development of others related to user interface and user characterization. As for R5, the network adopts a layered architecture, which starts from more general, domain-independent concepts, to concepts that cross the domain until reaching specific concepts of the HCI sub-domains (presented in Chapter 4). To satisfy R6, we have defined evolution mechanisms, which establish procedures for the integration and alignment of ontologies, enabling to integrate new and existing ontologies to the network. As for R7, the developed networked ontologies must meet the characteristics of "beautiful ontologies" (set of ontology assessment criteria that aim to achieve high quality ontologies) (D'Aquin; Gangemi, 2011). To meet R8, the network is integrated with SEON (Software Engineering Ontology Network) (Ruy et al., 2016), an ON that contains ontologies covering several aspects of the Software Engineering domain (presented in Chapter 4). Concerning R9, we have used HCI-ON in some applications to solve interoperability and knowledge-related problems (presented in Chapter 7). Finally, as for R10, HCI-ON conceptualization can be used as a whole or partially, according to the problem to be solved (discussed in Chapter 4).

To evaluate HCI-ON considering C1, the networked ontologies were validated by instantiating their concepts with data from real-world scenarios. In terms of C2, the mechanisms were used by the doctorate candidate herself and by other ontology engineers to evolve the ON, which allows to evaluate if the mechanisms were appropriate. Last, as for C3, the evaluation was carried out in the context of the use of HCI-ON to develop knowledge-based and ontology-based systems.

Finally, the ***Rigor Cycle*** refers to using and generating knowledge. Rigor is achieved by appropriately using foundations and methodologies from a knowledge base grounding the research, and adding knowledge generated by the research to contribute to the growing knowledge base (Hevner, 2007). As foundations, we adopted HCI relevant literature, including standards, theories, models and others. We also used knowledge of ontologies, ontology networks and secondary studies. The main contribution for the knowledge base is HCI-ON (Costa et al., 2020).

The main stakeholders that can benefit from HCI-ON include HCI researchers, professors and professionals that work with HCI or its intersection with other areas. They can use HCI-ON conceptualization to better understand the HCI domain (e.g., HCI phenomenon, human-centred design and evaluation, user interface types and elements, modalities of interaction, user characterization, etc.), contributing to teaching, learning and communication purposes. Ontology engineers, in turn, can use HCI-ON as a whole or extracts of it to build other ontologies. Software engineers (and related professionals) interested in producing knowledge-based or interoperability solutions using the networked ontologies, will also benefit from HCI-ON.

As secondary contributions we have (i) the secondary study that investigated HCI ontologies, which provides a panorama of ontologies addressing HCI aspects and how they have been developed and used to solve problems in the HCI domain (Costa; Barcellos; Falbo, 2021); and (ii) ontology-based solutions to knowledge-related problems. Concerning (ii), the result produced in this work consists of UXON (User eXperience Ontology Network-based system) (Manso, 2022), a system that uses an extract of HCI-ON to support user experience evaluation. In the context of related works carried out in the same research group, HCI-ON was used in the development of a system that supports knowledge recording and sharing to aid in human-centred design (Castro et al., 2021) and also in the development of an adaptive interface of a social network for users with different degrees of color blindness (Scalser, 2022).

During this research, we have elaborated papers presenting some of the results produced so far. The following papers were published in conferences:

- COSTA, S. D., BARCELLOS, M. P., FALBO, R. de A., & CASTRO, M. V. H. B. Towards an Ontology Network on Human-Computer Interaction. In G. DOBBIE, G. et al (Ed.), *Proceedings of the 39th International Conference on Conceptual Modeling*. Cham: Springer International Publishing, 2020. p. 331–341. ISBN 978-3-030-62522-1. DOI: 10.1007/978-3-030-62522-1_24.

- CASTRO, M. V. H. B., COSTA, S. D., BARCELLOS, M. P., & FALBO, R. de A. Knowledge Management in Human-Computer Interaction Design: A Mapping Study. In: *Proceedings of the XXIII Iberoamerican Conference on Software Engineering*, CIbSE 2020, Curitiba, Brazil, November. Curitiba, Brazil: [s.n.], 2020. p. 9–13.

- CASTRO, M. V. H. B.; BARCELLOS, M. P.; FALBO, R. d. A; COSTA, S. D. Using Ontologies to aid Knowledge Sharing in HCI Design. In: *Proceedings of the XX Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2021. p. 1–7. ISBN 9781450386173. DOI: 10.1145/3472301.3484327.

The following papers were published in journals:

- COSTA, S. D.; BARCELLOS, M. P.; FALBO, R. d. A. Ontologies in human–computer interaction: A systematic literature review. *Applied Ontology*, p. 1–32, oct 2021. ISSN 18758533. DOI: 10.3233/AO-210255.

- COSTA, S. D., BARCELLOS, M. P., FALBO, R. de A., CONTE, T., & OLIVEIRA, K. M. de. A Core Ontology on the Human-Computer Interaction phenomenon. *Data & Knowledge Engineering*, v. 138, p. 101977, mar 2022. DOI:10.1016/j.datak.2021.101977.

- CASTRO, M. V. H. B., COSTA, S. D., BARCELLOS, M. P., & FALBO, R. de A. Investigating Knowledge Management in Human-Computer Interaction Design *Journal of Software Engineering Research and Development*, v. 9, n. 1, p. 20, 2022.

1.5 Organization of this Thesis

This chapter presented the Introduction of this thesis, involving the general aspects, namely: the context and motivation for this research, the research hypothesis and objectives and the adopted methodological approach. The next chapters are organized as follows:

- **Chapter 2. Background: HCI, Ontologies and Ontology Network:** presents the foundations required for grounding the ideas of this research. The content covers an introduction to HCI and its main aspects related to this work, and the most relevant ontological notions applied along the work, including ontology classification, ontology networks. An introduction to the Unified Foundational Ontology (UFO) (Guizzardi, 2005; Guizzardi; Falbo; Guizzardi, 2008; Guizzardi et al., 2013; Guizzardi et al., 2015; Guizzardi et al., 2018) and to the Software Engineering Ontology Network (SEON) (Ruy et al., 2016) portions used in this work is also presented in this chapter.

- **Chapter 3. A Secondary Study on HCI Ontologies:** presents an investigation, which associates systematic literature review and systematic mapping methods, conducted to investigate ontologies in HCI, aiming at providing a panorama of how ontologies have been used in the HCI domain and how they have been developed. The findings and results are presented and discussed.

- **Chapter 4. A Human-Computer Interaction Ontology Network:** presents HCI-ON, the Human-Computer Interaction Ontology Network proposed in this work, and discusses: HCI-ON premises and architecture; the integration of HCI-ON and SEON; the mechanisms to support HCI-ON evolution; the application of HCI-ON evolution mechanisms to support including new domain networked ontologies and to favor semantic interoperability initiatives. Moreover, this chapter also introduces the HCI-ON specification.

- **Chapter 5. A Human-Computer Interaction Ontology:** presents HCIO, the Human-Computer Interaction Ontology, which addresses the interaction phenomenon and its components, namely: user and interactive computer system. The chapter also presents two real cases of HCI phenomenon that are used to demonstrate that HCIO is able to represent real-world situations; discusses how HCIO was evaluated and its use as a core ontology of HCI-ON.

- **Chapter 6. A Human-Computer Interaction Evaluation Ontology:** presents HCIEO, the Human-Computer Interaction Evaluation Ontology, which addresses user-centred evaluation and it is a domain ontology of HCI-ON. The chapter presents HCIEO conceptualization and evaluation.

- **Chapter 7. HCI-ON Applications:** presents the use of HCI-ON to develop UXON (User eXperience Ontology Network-based system) (Manso, 2022) and the studies carried out to evaluate it from the users' perspective (aiming to verify if the use of HCI-ON helped producing a useful solution) and from the developer's perspective (aiming to verify if the use of HCI-ON helped developing the solution). Other two systems developed based on HCI-ON extracts are also briefly presented, namely: KTID (Knowledge Tool for Interaction Design) (Castro et al., 2021; Sessa, 2021) and SNOPI (Social Network with Ontology-based adaPtive Interface) (Scalser, 2022).

- **Chapter 8. Final Considerations:** summarizes the main ideas discussed in this thesis, addressing the research contributions and the impacts of this work, the current limitations, and, finally, perspectives of future work.

- **Apendix A: Artifacts used in UXON's Evaluation – User Perspective:** presents the instruments used for the questionnaires applied in the empirical studies.

- **Apendix B: Artifacts used in UXON's Evaluation – Developer Perspective:** presents the instruments used for the interview applied in the empirical studies.

2 Background: HCI, Ontologies and Ontology Network

This chapter presents the background for this work. It is related to the Rigor cycle (Figure 1.1), more specifically to the use of foundations from a knowledge base to ground this research. Section 2.1 concerns HCI and its main aspects related to this work. Section 2.2 regards ontologies, discussing its basic notions and classifications. It also presents the Unified Foundational Ontology (UFO) (Guizzardi, 2005) portion relevant to this work. Section 2.3 addresses ontology networks and presents the Software Engineering Ontology Network (SEON) (Ruy et al., 2016) portion used in this work. Last, Section 2.4 closes the chapter.

2.1 Human-Computer Interaction

The Human-Computer Interaction term has surfaced in the 1980s and is currently extremely in evidence with the large use of interactive systems supporting daily activities and the advent of new technologies (mobiles, tablets, and so on). HCI is interested in the design, implementation and evaluation of interactive computer systems for human use, along with the phenomena related to this use(Preece; Sharp; Rogers, 2015; Carroll, 2014). It has five study objects: (i) the nature of human-computer interaction, (ii) the use of interactive systems situated in context, (iii) human characteristics, (iv) the architecture of computer systems and the user interface, and (v) development processes with a focus on use (Barbosa; Silva, 2010; Hewett et al., 1992).

HCI is a multidisciplinary knowledge area that aggregates a vast and multifaceted community, bound by the evolving concept of usability, and the integrating commitment to value human activity and experience as the primary driver in technology (Carroll, 2014). It is connected to other research areas, involving knowledge from multiple areas, such as ergonomics, cognitive science, user experience, human factors, among other (Sutcliffe, 2014; Carroll, 2014; Preece; Sharp; Rogers, 2015).

The interest in interactive systems and their impact on people's life has promoted the study and practice of usability, a key aspect related to user efficiency and satisfaction when interacting with the computer. HCI is also concerned with other important qualities characteristics such as user experience, accessibility and communicability. The HCI discipline aims to promote the study and practice of usability and related characteristics seeking to analyze the impact that technology has on people's lives and create software and other technologies that people will want to use, will be able to use and will be useful when used (Carroll, 1997). It is responsible for the analysis, design, implementation and evaluation of interactive computer systems for human use (Preece; Sharp; Rogers, 2015).

2.1.1 Human-Computer Interaction Phenomenon

"Interaction, simply, is the core of HCI." (Hornbæk; Oulasvirta, 2017). HCI has evolved since the 1980s[1] through various terminologies, classifications and studies. An important study is the definition of the three paradigms to explain the HCI phenomenon proposed by (Harrison; Tatar; Sengers, 2007). The first paradigm sees interaction as man-machine coupling, aims at optimizing the fit between man and machine, and combines engineering and human factors. The second focuses on cognitive science and adopts the metaphor of mind and computer as coupled information processors, and aims at optimizing accuracy and efficiency of information transfer. The third sees interaction as phenomenologically situated, has in its center the meaning and meaning construction, and aims at supporting situated actions in the world.

Dix et al. (2004) consider the communication between user and interactive computer system the interaction itself. User and system are, thus, participants in the interaction. Briefly, a human-computer interaction is the communication process that occurs during the use of an interactive computer system and that involves user actions on the system interface (user input) and user interpretations of the system responses (system output) revealed through the user interface (Figure 2.1). The user interface includes all parts of the system that a user has contact with, physical, perceptually or conceptually (Benyon, 2010). Interactive computer systems aid in goals achievement by supporting the accomplishment of tasks in some application domain or context of use where users interact with the system through its interface.

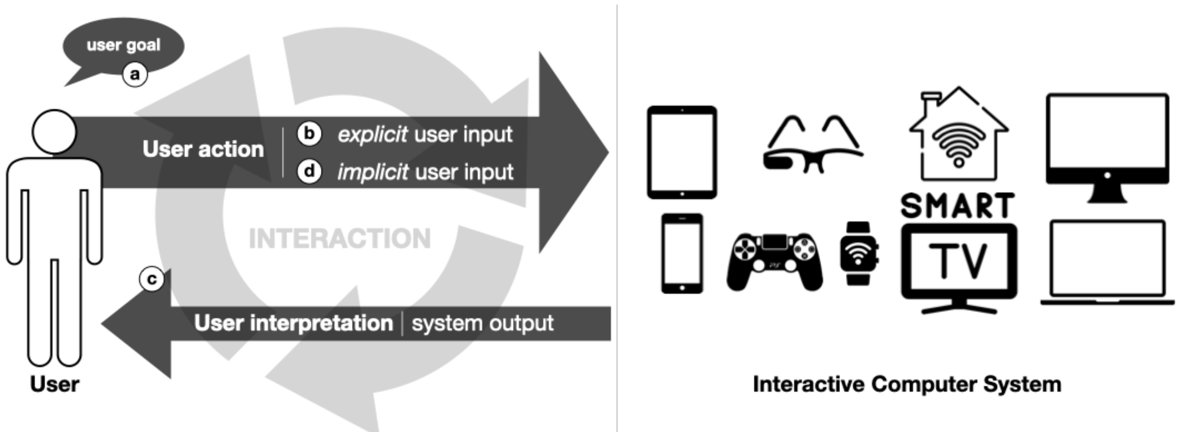


Figure 2.1 – Human-Computer Interaction: (a) user goal triggering the interaction, (b) user action, explicit user input, (c) system output (triggering the interaction or not) and user interpretation, (d) user action (does not rely on the user's intentionality), implicit user input

According to (Norman, 2013), the interaction cycle can start from the top, in a *goal-driven behavior* (Figure 2.1, (a)), where the user first establishes a goal to be achieved and then goes through user actions to accomplish the goal. In Figure 2.1, (a) together with (b) represents

[1] The HCI term started to be largely used after the publication of the book entitled "The Psychology of Human–Computer Interaction" by Stuart K. Card, Thomas P. Morton and Allen Newel, in 1983.

that the interaction starts with the goal establishment and a user action that triggers the interaction cycle. For example, Mary has the goal of sending an email to David (Figure 2.1, (a)). Thus, she types a message in her laptop (user input) and sends it to David (Figure 2.1, (b)). The email system shows a notification that the email was sent (system output) and Mary interprets that she has achieved her goal (Figure 2.1, (c)). The interaction cycle can also start from the bottom, in a *data-driven* or *event-driven behavior*, triggered by some event in the world (e.g., an event caused by an interactive computer system) and then can go through user actions (Figure 2.1, (c) when the system output triggers the interaction). In the previous example, consider that David (who was not expecting to receive an email from Mary at that moment) is notified by his smartphone (system output) that he has received an email (Figure 2.1, (c)). He interacts with his smartphone (inputting data in it and interpreting its output) due to the system notification (Figure 2.1, (b)). After that, he can start a new interaction cycle, now in a *goal-driven behavior* (starting from Figure 2.1, (a)), motivated by his goal of answering Mary's email.

The aforementioned perspective refers to *traditional* and *explicit human-interaction* (Schmidt, 2000), i.e., the user explicitly enters data through the user interface (explicit input) and the interactive system, in turn, uses output devices to deliver information (Schmidt, 2005). In Mary's example, she types the message because of her goal and, when she is typing the message (explicit input), she is aware of the changes occurring in the user interface (system output).

Some interactions, said *implicit human interaction*, are more transparent, natural and not explicit. They are performed by the user that is not aimed to interact with the interactive system, but the system understands the user actions as inputs (Schmidt, 2000). For example, David wears a smartwatch and is lying down reading a book. The smartwatch constantly captures physiological information (e.g, David's pressure and heart rate), i.e., David unintentionally provides inputs to the system (Figure 2.1, (d)). Suddenly, the wind blows through the window and David gets up very fast and runs to hold a door that would slam. The smartwatch notifies David (Figure 2.1, (c) system output) that his heart rate increased faster than usual in the last seconds. David then interprets that it was because he got up very quickly and ran (Figure 2.1, (c) user interpretation).

2.1.2 Human-centred Design and Evaluation of Interactive Systems

As previously said, HCI is interested in the design, implementation and evaluation of interactive systems for human use, as well as in the phenomena related to that use. It is strongly related to Software Engineering (SE) – both are design disciplines that share the common goal of developing interactive systems. However, they employ different perspectives of what is important in an interactive system, in its use and in its development. Therefore, each area gives rise to its own methods, techniques and processes (Barbosa; Silva, 2010; Sutcliffe, 2014). SE seeks systematic approaches with linear "specify-design-implement" processes. The HCI, in turn, is less method-oriented. It prioritizes design and evaluation over implementation and emphasizes the use of the system, focusing on the context of use, the user's goals and ways of achieving them, the user characteristics and possible forms of interaction in the user interface. Thus, HCI works to provide adequate support to the user when performing its activities and in the achievement of its goals (Barbosa; Silva, 2010; Sutcliffe, 2014).

Quality attributes that HCI and SE concern with also differ partially. SE aims at attributes such as availability, integrity, robustness, maintainability and recoverability. Recently, SE has adopted the term quality in use to represent the ergonomic concept of usability in SE in order to integrate usability with Engineering and Software Quality standards (Barbosa; Silva, 2010; Bevan, 2006; Sutcliffe, 2014). HCI, in turn, aims at attributes such as usability, user experience, accessibility and communicability (Barbosa; Silva, 2010).

Usability is defined as the extent to which a system, product or service can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a context of specific use (ISO, 2019a). It addresses users' cognitive, perceptual and motor skills during interaction and involves a set of criteria - learnability, memorability, efficiency, safety and satisfaction - as a consequence of this use (Nielsen, 1993; Barbosa; Silva, 2010). User experience (UX) relates to the users' emotions and feelings. According to Hassenzahl and Tractinsky (2006), UX is the consequence of the interrelationship of a user's internal state, the characteristics of the designed system and the context (or the environment) within which the interaction occurs. Accessibility refers to the removal of barriers that prevent access to the interface and its interaction. Finally, communicability concerns the ability of the interface to communicate the design logic to the user (Barbosa; Silva, 2010).

2.1.2.1 Human-Centred Design

HCI Design aims to serve users and other stakeholders interested in the project. Unlike computer-centred design practiced in SE, HCI design is user-centred (Barbosa; Silva, 2010; Chammas; Quaresma; Mont'Alvão, 2015). User Centred Design (UCD) is a philosophy based on ergonomics, usability and human factors. It is based on the needs and interests of the user, which influences the form the product takes. It is a project-oriented approach that focuses on the use and development of interactive systems with an emphasis on making products usable and understandable. Its principles are: focus on the user (their characteristics, needs and objectives), observable metrics (user performance and reactions) and iterative design (repeat as many times as necessary) (Barbosa; Silva, 2010; Chammas; Quaresma; Mont'Alvão, 2015; Norman, 2002). The term Human-Centred Design (HCD) has been adopted in place of UCD, aiming to emphasize the impact to all stakeholders and not just those considered users (ISO, 2019a).

HCD proposes a set of activities to be followed during the product's life cycle from the initial stages of the project, even before the user interface design (Nielsen, 1993; Barbosa; Silva, 2010). Figure 2.2 shows the HCD process for internal systems defined by the ISO 9241 - part 210 (ISO, 2019a). This standard is intended for those responsible for planning and managing interactive systems projects through an iterative process (Bevan, 2006; ISO, 2019a).

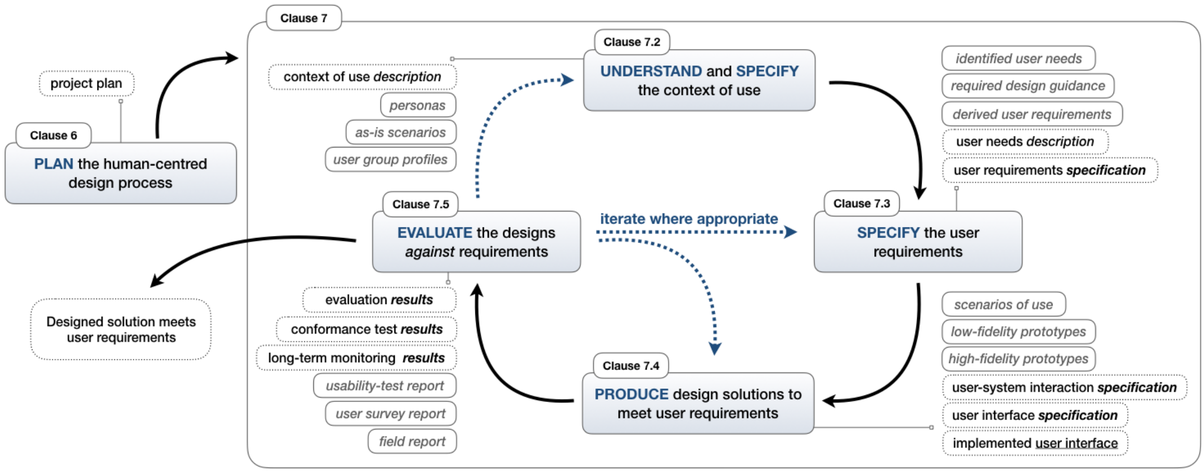


Figure 2.2 – Interdependency of human-centred design activities (adapted from (ISO, 2019a)).

Activities must start from the first stage of the project, the stage in which the initial product design is formulated, and can be incorporated into different SE design approaches (ISO, 2019a). ***Plan the HCD process*** aims at planning and integrating the HCD in all phases of the product life cycle: conception, analysis, design, implementation, testing and maintenance. This activity produces the *HCD project plan* as an artifact, which becomes part of the overall project plan (ISO, 2019a).

Understand and specify the context of use aims to study the users and anticipated uses of the product and describes in sufficient detail the context of use in the form of a document (*context of use description*), which must be indicated in the specification of user requirements in order to identify the conditions under which the requirements apply (ISO, 2019a). Such a document should address the groups of users, their characteristics, goals and tasks, as well as the system environment. Examples of artifacts produced in this activity are *personas*, *as-is scenarios* and *user group profiles*.

The ***requirements specification*** activity aims to identify the user's needs and specify functional requirements and other types of requirements for the product. In HCD, requirements specification activity is extended to create explicit statements of user requirements in relation to the context of intended use and system goals. This activity typically produces two artifacts, *user needs description* and *user requirements specification*. The former, regards the user and other stakeholders needs, taking into account the context of use. It should include what users need to achieve (rather than how to achieve it) and any constraints imposed by the context of use. The latter should include the intended context of use, requirements derived from user needs and context of use, requirements arising from relevant ergonomics and user interface knowledge, usability requirements and objectives, among others. Examples of these artifacts are *identified user needs*, *required design guidance* and *derived user requirements* (ISO, 2019a).

Produce design solutions aims to achieve the best user experience and includes sub-activities such as: design user tasks, user-system interaction and the user interface; make the most concrete solutions applying scenarios, simulations, prototypes and mock-ups; change the solutions due to the response of the evaluation and user feedback; and communicate the solution. This activity produces the following artifacts: *user-system interaction specification* (e.g., appropriate allocation of functions to be achieved, i.e., the division of system tasks into those performed by humans and those performed by technology - designing the interaction involves deciding how users will accomplish tasks with the system rather than describing what the system looks like), *user interface specification* (a detailed design of the user interface should take into account the ergonomics and user interface knowledge, standards and guidelines) and *implemented user interface* (enables designers to communicate the proposed design to users and other stakeholders to obtain feedback). Examples of these artifacts are *scenarios of use*, *low-fidelity prototypes* and *high-fidelity prototypes* (ISO, 2019a).

Evaluate the designs is based on the user's evaluation. It must be performed during all previous activities and can employ user-centred assessment approaches, such as user-based testing and inspection-based assessment, in order to obtain a design solution that meets user requirements and to better understanding of user needs. User-centred evaluation can be used to collect new information about the user, provide feedback of the design solution from the user's perspective, assess whether user requirements have been achieved and establish baselines or make comparisons between designs. Furthermore, to obtain valid results, the evaluation should be carried out by experienced evaluators, and should use appropriate methods (user-based or inspection-based). This activity aims to obtain early feedback to improve the product and to validate that the user requirements have been satisfied, producing the following artifacts: *evaluation results*, *conformance test results* and *long-term monitoring results*. Examples of these produced artifacts are *usability-test report*, *user survey report* and *field report* (ISO, 2019a).

2.1.2.2 Evaluation of Interactive Systems

According to Hassan and Galal-Edeen (2017), the success of an interactive system for human use does not depend only on its ability to provide functionality related to the users' goals and needs, but also on the experience provoked in the users while interacting with the system.

HCI is the area that values the quality of use (e.g., usability, accessibility, communicability, user experience) of interactive system intended for human use (Hassan; Galal-Edeen, 2017). The quality of use is related to characteristics of the interaction and of the interactive system interface (Barbosa et al., 2021). Therefore, HCI develops and applies evaluation methods that aim to assess the qualities of use of these systems and the phenomena related to this use (Preece; Sharp; Rogers, 2019; Carroll, 2014).

Among the existing qualities of use, usability and user experience are the main determinants of the system quality and success or failure indicators of this system (Hassan; Galal-Edeen, 2017). Hassan and Galal-Edeen (2017) argue that usability is a subset of the user experience and that both concepts complement each other.

In the literature, there is a diversity of methods that can be applied to assess each of the quality of use mentioned above. Also, different authors categorize them in different ways. For example, Preece, Sharp and Rogers (2019) classify the evaluation into three categories: controlled settings directly involving users; natural settings involving users; and any setting not directly involving users. Barbosa et al. (2021) consider two categories: HCI evaluation through inspection; and evaluation in HCI through observation. Petrie and Bevan (2009) categorize methods to evaluate the usability, accessibility and user experience of eSystems[2] in: (i) automated checking of conformance to guidelines and standards; (ii) evaluations conducted by experts; (iii) evaluations using models and simulations; (iv) evaluation with users or potential users; and (v) evaluation of data collected during usage. Along with this classification, the authors usually provide information on when to use and why conduct each of the evaluation methods. This information is crucial (and should be used during the design process presented in Section 2.1.2.1) to guide the evaluator to be assertive in choosing the correct evaluation method for what he/she wants to evaluate.

For example, in (ii), it is recommended: its use as soon as initial prototypes are available; to be conducted when there is no time or users available to carry out the evaluation, among others (Petrie; Bevan, 2009). Thus, in a certain scenario, in which good usability of the interactive system is desired, heuristic evaluation (Nielsen, 1994a; Nielsen, Jakob, 2020) may be the most appropriate choice. It involves a small group of evaluators (there is no user participation) who examine the system's user interface (or prototype) according to usability principles (recognized as "heuristics" (Nielsen, Jakob, 2020)) and judge its compliance with these principles (Nielsen, 1994a).

In (iv) there are methods in which the evaluation of usability and user experience are based on user observation (Petrie; Bevan, 2009), which allows recording of interaction data and measurement of collected data (Barbosa et al., 2021) to reach conclusions about these characteristics of quality of use. Observation ensures that the user is not interrupted during their interaction and experience (Preece; Sharp; Rogers, 2019).

Furthermore, observation methods can be classified as: direct (iv, user-based evaluation), when data is directly recorded by the evaluator observing the user; and indirect (v, data collected during usage), when the data is recorded by the system itself during its use (interaction logging), that is, indirectly recorded, and does not require the presence of the evaluator during the evaluation (Barbosa et al., 2021; Marques et al., 2020; Preece; Sharp; Rogers, 2019; Petrie; Bevan, 2009). In both, data from interactions and situations that may occur while the user interacts with the system being evaluated are recorded and analyzed (or measured) and allows identifying problems that occurred during their user experience (Barbosa et al., 2021; Marques et al., 2020; Preece;

[2] interactive electronic products, services and environments (eSystems) (Petrie; Bevan, 2009).

Sharp; Rogers, 2019).

The evaluation of interactive systems with a focus on user experience leads the evaluator to make an analysis of computer interactive systems and user behavior that allows the identification of errors in the interaction and in the interface that hinder the user experience (Barbosa et al., 2021; Marques et al., 2020; Menezes; Nonnecke, 2014). Direct observation is best when a small group of users is involved. On the other hand, when it is necessary to observe the behavior of many users, indirect observation becomes more appropriate. For example, one of the known methods in user-based evaluation is Usability Testing (Moran, 2019), and in indirect evaluation it is A/B Testing (Preece; Sharp; Rogers, 2019). Both involve metrics and measurements of the data collected and can be complemented with interviews and questionnaires applied to users after using the system (Preece; Sharp; Rogers, 2019; Petrie; Bevan, 2009).

A metric[3] allows characterizing a particular entity by quantifying its properties (Barcellos; Falbo; Frauches, 2014). Thus, metrics related to user experience quantitatively describe some perspective of this experience (Albert; Tullis, 2013). Thus, they show, based on quantitative values, some aspect of the interaction between the user and the system, such as, for example, effectiveness (ability to perform a task); efficiency (the amount of effort used to complete the task); or satisfaction (how happy the user was with their experience while performing the task). A striking feature among metrics related to user experience is that they measure something related to human beings and their behavior (Albert; Tullis, 2013).

2.2 Ontologies

The term ontology has been used in philosophy to refer both to a philosophical discipline, and as a domain-independent system of categories. In Computer Science, we refer to an ontology as a special kind of information object or computational artifact (Guarino; Oberle; Staab, 2009).

Originally defined as an "*explicit specification of a conceptualization*" (Gruber, 1993) the ontology definition was adapted to a "*formal specification of a shared conceptualization*" (Borst, 1997). Moreover, both definitions were merged and adapted to express a shared view between several parties, a consensus rather than an individual view. Gruber (1995) stated that: "*an ontology is a formal, explicit specification of a shared conceptualization.*"

In the literature, ontologies have been classified considering diverse perspectives, such as according to their levels of generality, formality, applicability, etc. In this research, we are mainly interested in the classification criterion regarding *Generality Levels*. According to Scherp et al. (2011), ontologies can be classified into foundational, core, and domain generality levels. **Foundational ontologies** aim at modeling the very basic and general concepts and relations that make up the world (including domain-independent notions, such as *objects*, *events*, *participation* and *parthood*) (Guarino, 1998). They are generic across any area and are highly reusable in

[3] In this work, the terms *metric* and *measure* are adopted as synonymous.

different modeling scenarios (domain-independent) and represent the highest-level ontologies. **Core ontologies** provide a refinement to foundational ontologies by adding detailed concepts and relations in a specific area (such as service, process, organizational structure) that still spans across various (sub)domains. Core ontologies are situated in between foundational and domain ontologies and despite being more general than domain ontologies, are domain-dependent. Finally, **domain ontologies** describe knowledge that is specific for a particular domain, such as a soccer ontology (Guarino, 1998) and represent the lowest-level ontologies (e.g., an ontology about the anatomy of the human body). They can make use of/be based on foundational ontologies or core ontologies by specializing their concepts. Higher-level ontologies can be used to support the development of lower-level ontologies, i.e., foundational ontologies can be used as basis for building core and domain ontologies, and core ontologies can support the development of domain ontologies.

Falbo et al. (2013) argue that Scherp, Saathoff, Franz, & Staab (Scherp et al., 2011) classification should be perceived as a continuum, ranging from pure foundational ontologies (that are totally domain-independent) to domain ontologies (domain-dependent). Thus, there can be different levels of generality in ontologies classified in a certain type. In this sense, considering the continuous nature of the aforementioned classification, some ontologies can be used for supporting the development of more specific ontologies even within the same level of generality. For instance, there are more general core ontologies, such as UFO-S (Nardi; Falbo; Almeida, 2013), which addresses services in general, and more specific core ontologies, such as the Software Process Ontology (SPO, presented in Section 2.3.1.2) (Bringuente; Falbo; Guizzardi, 2011), which addresses core concepts about software and system in the Software Engineering area and spans across several sub-domains in that area, such as software measurement, software design process, software test process and so on.

Figure 2.3 illustrates the view of ontology generality levels as a continuum. The dashed arrows show the grounding dependencies between the ontologies in different levels.

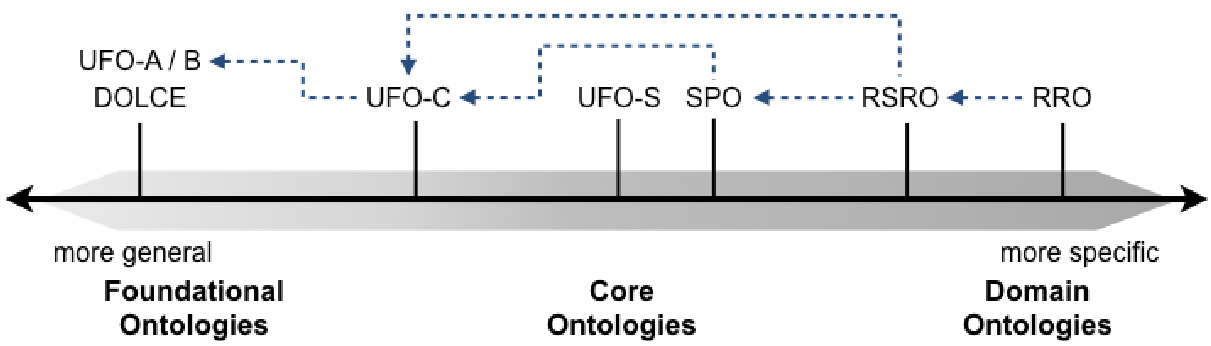


Figure 2.3 – Ontology Generality Levels as a Continuum (adapted from (Falbo et al., 2013; Ruy, 2017)).

In the figure, DOLCE (Borgo; Masolo, 2009; Gangemi et al., 2002) and the Unified

Foundational Ontology - UFO (Guizzardi, 2005) are foundational ontologies. UFO-C (an ontology of social entities) is still considered to be a foundational ontology despite being more specific than UFO-A (an ontology of endurants) (Guizzardi, 2005; Guizzardi et al., 2018) and UFO-B (an ontology of events) (Guizzardi; Falbo; Guizzardi, 2008; Guizzardi et al., 2013). The *Reference Software Requirements Ontology* (RSRO, presented in Section 2.3.1.4) (Ruy et al., 2016) and *Runtime Requirements Ontology* (RRO) (Duarte et al., 2016) are domain ontologies, although RRO addresses an even more specific domain than RSRO.

Another important distinction concerns the ontology intended application and differentiates ontologies as conceptual models, called **reference ontologies**, from ontologies as computational artifacts, called **operational ontologies** (Guizzardi, 2007). A reference ontology is a special kind of conceptual model, an engineering artifact, constructed with the goal of making the best possible description of the domain in reality, representing a model of consensus within a community, regardless of its computational properties (Guizzardi, 2007; Studer; Benjamins; Fensel, 1998). Once users have agreed on a common conceptualization, different operational versions (machine-readable ontologies) of a reference ontology can be implemented. Contrary to reference ontologies, operational ontologies are designed with the focus on guaranteeing desirable computational properties.Moreover, when developing a reference ontology, the focus is on expressivity of the representation and truthfulness to the domain being represented (domain appropriateness), even at the expenses of computational characteristics such as tractability and decidability (Guizzardi, 2007). Summing up, in the view employed here, a reference ontology is a particular kind of conceptual model, namely, a reference conceptual model capturing the shared consensus of a given community.

Using ontologies provides several benefits. Feilmayr and Wöß (2016) categorize them in technical-centred or user-centred. From a technical point of view, some benefits are: (i) enables communication and facilitates knowledge transfer between computational systems, between humans, and between humans and computational systems; (ii) enables computational inference, which is useful for deriving implicit facts; (iii) promotes knowledge organization, making domain assumptions explicit and sharing an understanding of the information structure; (iv) enables, on the one hand, reuse of domain knowledge and, on the other, integration of new knowledge built upon existing knowledge; (v) promotes standardization, contributing to establish a uniform language that enables protocols; (vi) provides identification, i.e., a unique identifier, uniquely identifies the meaning of concepts in a given domain of interest; (vii) clearly separates between an ontological schema and its instances; (viii) allows constant evolution, allowing agile schema management during application runtime, which is supported by the graph-based data model, in contrast to relational databases. From the user point of view, some benefits are: (i) enables knowledge sharing between users; (ii) makes domain assumptions explicit and thus offer knowledge base schemas for storing and retrieving information; (iii) provides a domain specification as a result of an intensive domain analysis; (iv) provides a clear separation of operational and domain knowledge; and (v) enables knowledge reuse.

In this work, the focus is on solving knowledge-related and interoperability problems in the HCI domain. Even with the support of methods and tools, it is still an activity essentially performed by humans. Thus, in this text, when referring to ontologies, we mean reference ontologies (except when we explicitly refer to operational ontologies). Regarding the generality levels, all of them are applied, as we will discuss in Chapter 4. Concerning the foundational ontology, we use the Unified Foundational Ontology UFO (Guizzardi, 2005) to provide the common ontological foundation to the proposed network. Hence, next we briefly present UFO and its fragment relevant to this work.

2.2.1 The Unified Foundational Ontology - UFO

UFO is a foundational ontology that has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Linguistics and Cognitive Psychology (Guizzardi et al., 2015). It is constituted by three main parts. UFO-A is an ontology of endurants (*objects*) (Guizzardi, 2005; Guizzardi et al., 2018), UFO-B, an ontology of events (*perdurants*) (Guizzardi; Falbo; Guizzardi, 2008; Guizzardi et al., 2013), and UFO-C (Guizzardi; Falbo; Guizzardi, 2008), an ontology of *social entities* built on the bottom of UFO-A and UFO-B. Figure 2.4 and 2.5 present the fragment of UFO used to ground this research. The description below is based mainly on (Guizzardi; Falbo; Guizzardi, 2008) and (Guizzardi et al., 2013).

In UFO, *Entities* are classified into Individuals and Universals. *Universals* (will be explored later) are patterns of features, which can be realized in a number of different individuals. *Individuals* (shown in Figure 2.4), in turn, are entities that exist in reality possessing a unique identity. In other words, universals are entities that are possibly instantiated (e.g., Person), while individuals are those that necessarily cannot be instantiated (e.g., John).

UFO makes another fundamental distinction between enduring and perduring individuals. *Endurants* are said to be wholly present whenever they are present (e.g., a person), i.e., they *are in time*, while *Perdurants (Events)* are individuals composed of temporal parts (e.g., a soccer game), i.e., they *happen in time* in the sense that they extend in time accumulating temporal parts. *Endurant Universals* are universals whose instances are endurants, while *Event Universals* are universals whose instances are events.

In UFO-A, there are three main categories of endurants: Substantials, Moments and Situations. *Substantials* are existentially independent individuals (such as a person, a house, a car etc.). *Moments*, in contrast, are individuals that can only exist in other individuals. Quite simply, moments denote properties, and thus, they can only exist in other individuals. In other words, moments are existentially dependent on other individuals. *Intrinsic Moments* are those moments that depend on one single individual (e.g., the color of a car, the temperature of a person). *Situations* are complex entities that are constituted by possibly many endurants (including other situations). A situation represents a state of affairs, i.e., a portion of reality that can be comprehended as a whole. “John being with fever and influenza” is an example of a situation.

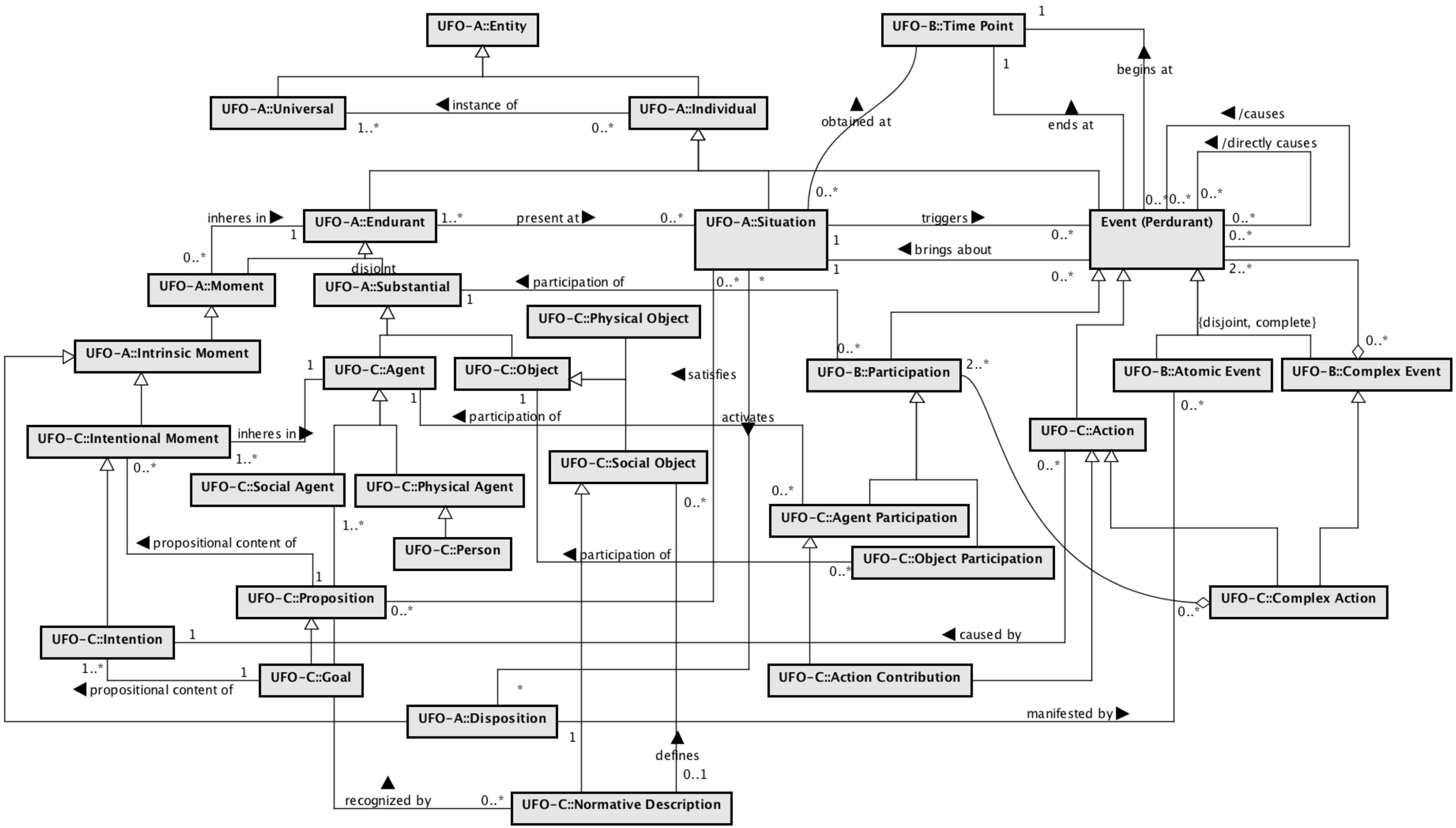


Figure 2.4 – UFO Fragment relevant to this research.

Endurants are present in the situations they constitute. For instance, both the substantial John, and the intrinsic moments m1 (John's Fever) and m2 (John's influenza) are present in the situation s1: "John being with fever and influenza". Finally, *Disposition* is a special kind of Intrinsic Moment, representing properties that are only manifested in particular situations and that can also fail to be manifested (e.g., the fragility of a glass, or the disposition of a magnet to attract metallic material). When manifested, they are manifested through the occurrence of events.

Regarding Perdurants (UFO-B), Events can be atomic or complex, depending on their mereological structure. Whilst *Atomic Events* have no proper parts, *Complex Events* are aggregations of at least two disjoint events. Events are ontologically dependent entities in the sense that they existentially depend on substantials to exist. The existential dependence of events on substantials provides for an orthogonal way of partitioning events. Besides the mereological decomposition of events, a complex event can be partitioned by separating each part of this event which is existentially dependent on each of its participants. The portion of an event which depends exclusively on a single substantial is said a *Participation* (e.g., the participation of a player in a soccer game). As an orthogonal way of partitioning events, participations can be atomic or complex.

Events are transformations from a portion of reality to another, i.e., they may change reality by changing the state of affairs from one situation to another. There are two possible relations between situations and events: (i) a situation *triggers* an event in the case that the event occurs because of the obtaining of the situation, and; (ii) an event *brings about* a situation in the case that the occurrence of the event results in the situation. Suppose that a situation s was brought about by an event e, and that s triggered another event e'. In this case, we can state that the occurrence of e *directly causes* the occurrence of e'. Moreover, if e' *directly causes* another event e", then we can state that the first event e *causes* the last event e". Both relations *directly causes* and *causes* are derived relations (and thus represented in the model in Figure 2.4 preceded by '/'), since they can be obtained by means of the intermediate situations.

UFO-C starts by making a distinction between agentive and non-agentive substantial individuals, termed *Agents* and *Objects*, respectively. Agents can be physical or social. *Person* is a subtype of *Physical Agent*. Agents are substantials that can bear special kinds of intrinsic moments named *Mental Moments*. Every mental moment has a *Proposition* as its propositional content, which refers to a class of situations referred by that mental moment. "Intending something" is a specific type of mental moment termed *Intention*. The propositional content of an Intention is a *Goal*. The relation between a mental moment and a situation is the following: Situations in reality can satisfy the propositional content of a mental moment (i.e., satisfy in the logical sense the proposition representing that propositional content). Thus, intentions can be fulfilled or frustrated. Intentions are desired situations for which the agent commits at pursuing. For this reason, intentions cause the agent to perform actions. *Actions* are intentional events, i.e., events with the specific purpose of satisfying (the propositional content of) some intention.

Participations can also be intentional (i.e., be themselves actions) or unintentional events. For example, the stabbing of Caesar by Brutus includes the intentional participation of Brutus and the unintentional participation of Caesar and the knife. Thus, it is not the case that every *Agent Participation* is considered an action, but only those intentional participations, which are said *Action Contributions*. Only agents (entities capable of bearing intentional moments) can perform actions. The participations of objects in events are always unintentional and are said *Object Participations*. Moreover, as *Agents*, *Objects* can be physical (*Physical Object*) or social (*Social Object*). *Physical Objects* can be a book, a knife, while *Social Objects* include a project and money. *Normative Descriptions* is a *Social Objects* that define one or more rules/norms recognized by at least one *Social Agent* and that can define other *Social Objects*. For example, the Brazilian Constitution (*Normative Description*) defines the Child and Adolescent Statute (*Social Object*, more specifically a *Normative Description*), and both are recognized by Brazilian citizens (*Social Agent*).

Figure 2.5 shows a fragment of UFO that addresses *Qualities*. This fragment is important to ground concepts related to the measurement of properties of entities (e.g., collection of values related to metrics aiming to evaluate usability of an interactive system).

The root concept of UFO is *Thing*, which is specialized into *Urelement*, *Abstract Thing* and *Set*. *Sets*, in turn, are entities that simply exist, without being explicitly created or destroyed. *Urelements* are all entities that are not *Sets* (e.g., the entities represented in Figure 2.4). *Intrinsic Moment Universals* are dependent on a single entity (e.g., Color). *Quality Universals* are *Intrinsic Moment Universals* and refer to the properties that characterize *Universals* (e.g., Weight and Height). They are *Intrinsic Moment Universals* (i.e., are dependent on a single entity (e.g., Color)) associated to *Quality Structures*, which can be understood as the set of all possible regions that delimits the space of values possible to be associated to a particular *Quality Universal*. For instance, the quality universal Weight is associated to a space of values that is a linear structure isomorphic to the positive half-line of the real numbers. The regions that compose a *Quality Structure* are called *Quality Regions*, and are regions that approximate qualia. A *Quale* is a perception of a *Quality* in a *Quality Structure* (e.g., whether the *Quality* is the height that characterizes a given person, the point in the *Quality Structure* that indicates the height of the person is the quale). *Quality Function* is a specialization of *Set* that maps instances of a *Quality Universal* to points in a *Quality Structure*.

Measurable Quality Universals are quality universals that can be objectively measured by cognitive agents or measurement devices, and it is possible to establish distances among their quality regions (e.g., Weight and Height). *Measurement Quality Structures* are structures that allow for objectively evaluating the distance between two values and verifying if the values are equal or not.

As said before, *Quality Regions* are regions that approximate qualia. A *Quale* is intrinsic to cognitive agents and therefore cannot be shared or communicated. In order to allow quale

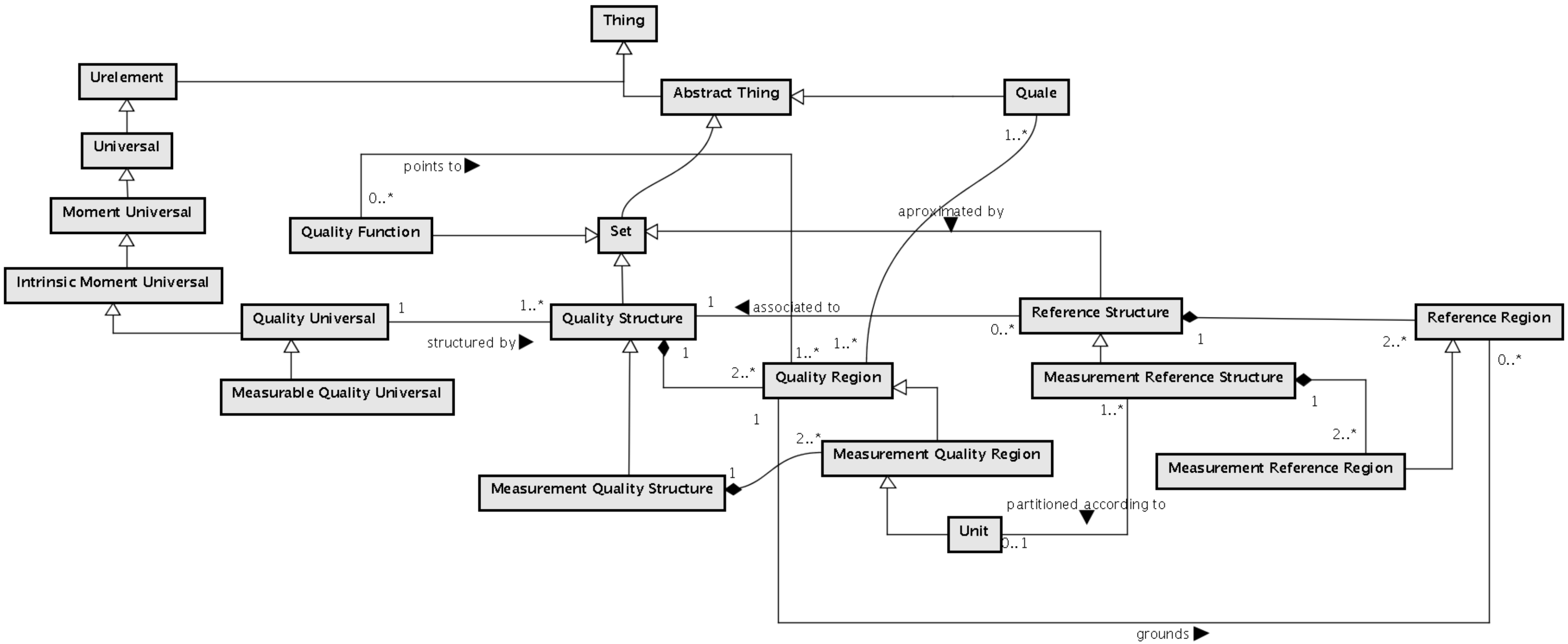


Figure 2.5 – UFO Fragment that addresses *Qualities* relevant to this research.

communication, it is necessary to use lexical elements (e.g., 1.86 can be the lexical element used to communicate the height of a person) associated to *Reference Regions* and *Reference Structures*. A *Reference Region* is an abstract entity based on a *Quality Region* that acts as a bridge between that region and the lexical elements used to communicate the approximated quale. A *Reference Structure*, in turn, is associated to a *Quality Structure* and is a set of *Reference Regions* grounded in *Quality Regions* of that *Quality Structure*. When the 'value' of a particular quality is being referred by lexical elements (e.g., 1.86), what is actually being referred is a quality region that most approximates the quale.

Finally, *Reference Structures* associated to *Measurement Quality Structures* are called *Measurement Reference Structures* and act like scales grounded by quality structures. They are composed by *Measurement Reference Regions* and can be partitioned in spaces with the same magnitude according to a *Unit*.

The UFO distinctions (such as these just described) have been used in diverse applications, such as providing ground for the development of core and domain ontologies and for analyzing conceptual models in a foundational light. The idea behind ontological analysis process basically consists in analyzing the concepts and definitions of a subject model and evaluating if they make sense according to the chosen semantic basis. The results range from identified problems and proposed solutions to the complete reengineering of that model. The employed basis use to be a formal and well accepted semantic reference. A foundational ontology, consistently providing the basic types that describe the world, is a good choice. Examples of the use of UFO in ontological analysis are (Guizzardi; Falbo; Guizzardi, 2008; Bringuente; Falbo; Guizzardi, 2011; Salamon et al., 2017).

2.3 Ontology Network

Ontologies are not stand-alone artifacts. They relate to other ontologies in ways that might affect their meaning, and are inherently distributed in a network of interlinked semantic resources (Ruy, 2017). An *Ontology Network* (ON) is a collection of ontologies, included in such a network, related together, through a variety of relationships, such as modularization, alignment and dependency, sharing concepts and relations with other ontologies (Suárez-Figueroa et al., 2012). Accordingly, a networked ontology is an ontology included in such a network, sharing relationships with a potentially large number of other ontologies (Suárez-Figueroa et al., 2012).

Specially for a complex domain, representing its knowledge as a single ontology results in a large and monolithic ontology that is hard to manipulate, use, and maintain (Suárez-Figueroa et al., 2012). On the other hand, representing each sub-domain in isolation is a costly task that leads to a very fragmented solution that is again hard to handle (Ruy et al., 2016). In such cases, building an ON is an adequate solution (Suárez-Figueroa et al., 2012).

ONs enable to establish a comprehensive conceptualization that provides a common understanding of the domain and can be used as a reference to solve semantic interoperability and knowledge problems related to the conceptualization as a whole or to extracts of it. Hence, integrating several ontologies into an ON provides a framework that can be explored to potentialize and increase the set of solutions in the universe of discourse addressed by the ON.

Developing an ontology network may require effort (particularly at the beginning, if existing ontologies need to be put in correspondency, merged, integrated, etc.). However, the effort is worth because the ontology network potentializes knowledge reuse and growth and, as a consequence, promotes more robust and comprehensive ontology-based solutions (Costa et al., 2020). Moreover, if ontologies are organized in an ON, when ontologies are needed in scenarios spanning different sub-domains, instead of spending effort to integrate several ontologies, one can just extract the ON portion to be used.

ON contributes to knowledge growth. It is like a living organism, i.e., it is constantly evolving. As the network is being populated, more and more common entities are defined and reused, in such a way that, for adding new networked ontologies, much of the effort and definitions have already been done. This mechanism makes the development of networked ontologies more productive (Ruy et al., 2016). New ontologies can be gradually added to it. Each new ontology contributes to the whole network. When a new ontology is added, it reuses elements from the networked ontologies. These, in turn, may be adapted to keep consistency and share the same semantics along the whole network.

Suárez-Figueroa et al. (2012) proposed the NeOn Methodology Framework, which provides guidance for engineering networked ontologies containing detailed processes, guidelines and different scenarios for collaboratively building networked ontologies (Ruy et al., 2016). One of the most common ways for two ontologies to relate is to be dependent on each other. Dependency relationships occur when, in order to define its own model, an ontology refers to concepts and relations defined in another ontology (i.e., an ontology reuses concepts from another). It can be done by using an OWL primitive (owl:imports) in operational ontologies, or by relating to or specializing a concept from other ontology in reference ontologies (Ruy et al., 2016).

Another manner to relate ontologies is aligning them. Alignment relationship is a way to put different models in correspondence by establishing equivalency mappings between entities from different ontologies (i.e., the same as, a generalization of, a specialization of) (Suárez-Figueroa et al., 2012).The main purpose of alignments is to ensure semantic interoperability, making it possible to merge ontologies in a meaningful way by representing information in one ontology in terms of the entities in another (Suárez-Figueroa et al., 2012).

In this thesis, due to the strong connection between IHC and Software Engineering, we reused concepts from SEON, a Software Engineering Ontology Network (Ruy et al., 2016). Next, we introduce SEON and its fragments relevant to this work.

2.3.1 Software Engineering Ontology Network - SEON

The Software Engineering Ontology Network (SEON) results from several efforts on building ontologies for the Software Engineering (SE) area (Ruy et al., 2016).

SEON is organized in a three-layer architecture. Briefly, at the foundational layer, there is the Unified Foundational Ontology (UFO, presented in 2.2.1). At the core layer, there are three core ontologies: the Software Process Ontology (SPO), the Core Ontology on Measurement (COM) and Enterprise Ontology (EO). Finally, the domain-specific layer encompasses twelve SE sub-domain ontologies (such as Reference Software Requirement Ontology (RSRO), Reference Ontology on Software Testing (ROoST), Design Process Ontology (DPO), among others. These represent the SEON version available at <http://dev.nemo.inf.ufes.br/seon>[4].

During the development of this work, some changes were made to Software Ontology (SwO) (Duarte et al., 2018). It incorporated concepts related to hardware equipment resources, originally defined in SPO, and to systems, was renamed as Software and System core Ontology (SysSwO) and moved from the domain to the core layer (that is, the ontology scope was increased and it now addresses core aspects related to software and systems). Next, we present extracts of SEON that are relevant to this research. They contain concepts reused to develop networked ontologies of HCI-ON.

2.3.1.1 System and Software Ontology - SysSwO

SysSwO is a core reference ontology about the nature of system and software, including, software artifacts, software constitution, software execution, computer system and hardware equipment. It aims to capture the complex artifactual nature of system and software, explaining what a computer system is, how software is materialized and executed inside a computer system, producing results that go beyond the limits of the machine, directly affecting the real world (Bringuente; Falbo; Guizzardi, 2011; Duarte et al., 2018). In this work, we address interactive computer systems and how user interacts with them. Thus, we reuse the general conceptualization about system and software to define relevant HCI concepts (e.g., through specializations). Like all SEON core and domain ontologies, SysSwO is grounded in UFO. Figure 2.6 presents the SysSwO fragment that was reused in this research. In the sequel, we describe the main concepts reused and how they are grounded in UFO (blue arrows in Figure 2.6). In the text, concepts from SysSwO are written in <u>*underline italics*</u> and concepts from UFO are written in *italics*. In the figure, UFO concepts are shown in grey while SysSwO concepts are shown in green.

<u>*Computer System*</u> is an *Object* combining hardware and software. Concerning hardware, a <u>*Computer System*</u> is composed of a set of <u>*Computer Machines*</u>. <u>*Computer Machine*</u> is a <u>*Hardware Equipment*</u> which other <u>*Hardware Equipment*</u> connect to it. <u>*Hardware Equipment*</u> is a physical *Object* used for running software programs or to support some software process

[4] Accessed on March 4th, 2021.

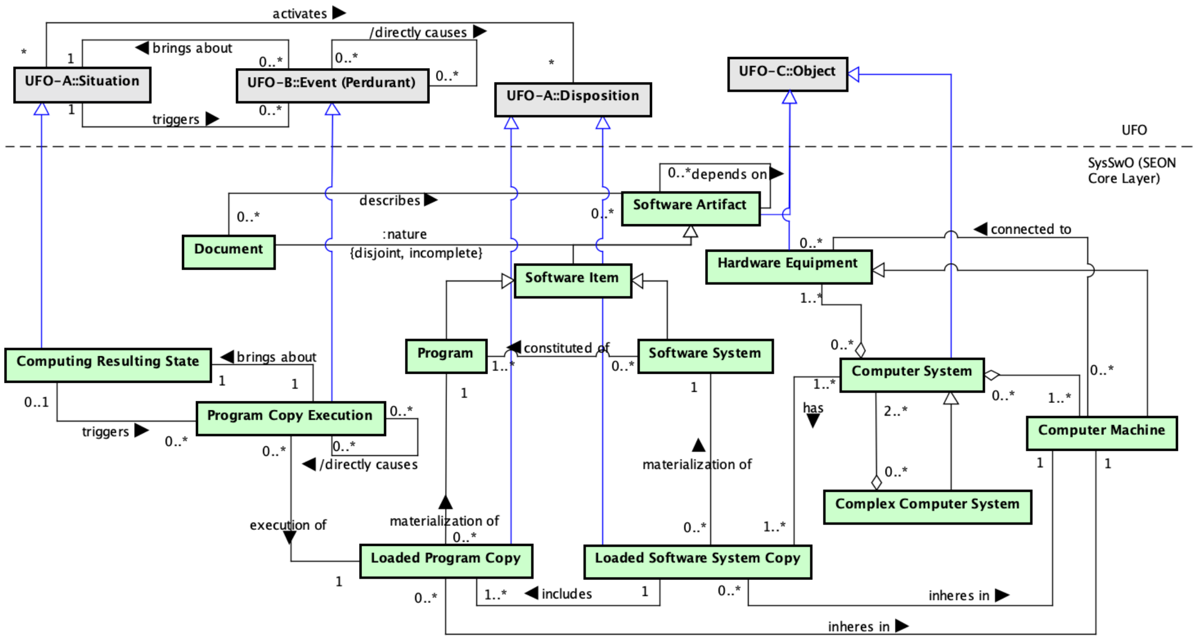


Figure 2.6 – SysSwO fragment relevant for this research.

activity (e.g., a printer, a router, a mouse, a keyboard).

Regarding software, a *Computer System* has a set of copies of software systems that are installed and loaded in the *Computer Machines* that comprise the *Computer System*. A copy of a software system installed and loaded in a *Computer Machine* is said a *Loaded Software System Copy* and it is a *Disposition* inhering in the *Computer Machine* where it is loaded. A *Loaded Software System Copy* is the materialization of a *Software System*, which is a subtype of *Software Software Item*. A *Item*, in turn, is a piece of software produced during the software development process, not considered a complete software product (e.g., a program). *Software Item* is a subtype of *Software Artifact* (can depends on another *Software Artifact*), which is an *Object* intentionally made to serve a given purpose in the context of a software project or organization. Moreover, *Software Artifact* can be described by a *Document* which, in turn, is subtype of *Software Artifact* and can be any written or pictorial information related to the software development (e.g., a requirement document, a specification, a report).

A *Software System* is constituted of *Programs* (also a subtype of *Software Item*). Analogously to a *Software System*, to run a *Program*, one must have a copy of the program loaded in a *Computer Machine* (*Loaded Program Copy*) and execute it. *Program Copy Execution* is then an *Event* that brings about a particular *Situation* (the post-state of the event termed as a *Computing Resulting State*) resulting from the *Program Copy Execution*. A *Computing Resulting State* (*Situation*) can trigger other *Program Copy Execution* (*Event*). The *Program Copy Execution* (*pce1*) that brought about a *Computing Resulting State* that triggered a second *Program Copy Execution* (*pce2*) is said to directly cause it (*pce1* directly causes *pce2*).

Finally, a *Computer System* can be composed of others *Computer Systems*. In this case, it is said a *Complex Computer System*.

2.3.1.2 Software Process Ontology - SPO

The Software Process Ontology (SPO) (Bringuente; Falbo; Guizzardi, 2011) is a core reference ontology that aims at establishing a common conceptualization of the Software Process domain, including processes, activities, resources, people and procedures. In this work, we address stakeholders responsible to perform a user-centred evaluation of an interactive computer systems and procedures to be followed during an evaluation. Thus, we reuse the general conceptualization about software process assets to define relevant HCI concepts (e.g., through specializations). Figure 2.7 presents the SPO fragment that was reused in this work. In the text, concepts from SPO are written in ***bold italics*** and concepts from UFO are written in *italics*. In the figure, SPO concepts are shown in yellow.

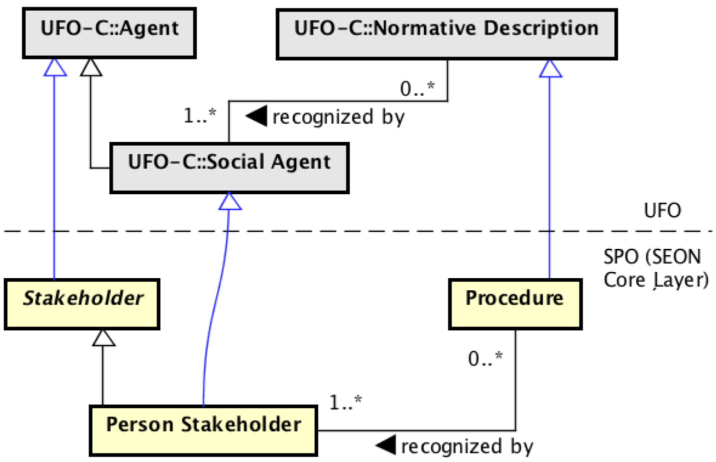


Figure 2.7 – SPO fragment relevant for this research.

A ***Stakeholder*** represents *Agents* (person, team or organization) interested or affected by the software process activities or results. Moreover, considering the agent nature, ***Stakeholders*** can be classified as ***Person Stakeholder*** (such as a hired Programmer, an external Instructor, or a User). As being *Social Agent*, ***Person Stakeholder*** recognizes ***Procedures*** which are *Normative Descriptions* prescribing a defined way for performing the activities that adopt it. ***Procedures*** relate to methods, techniques and document templates adopted by software activities.

2.3.1.3 Core Ontology on Measurement - COM

The Core Ontology on Measurement (COM) (Barcellos; Falbo; Frauches, 2014) is a core reference ontology that aims at establishing a central conceptualization related to the measurement, that is, the conceptualization that is independent of the domain in which the measurement is applied. In this work, we address interactive computer system evaluation which involves the measurement of its quality characteristics. Thus, we reuse the general conceptualization about measurement to define relevant user-centred evaluation concepts (e.g., through specializations).

Figure 2.9 presents the COM fragment that was reused in this work. In the text, concepts from COM are written in ***bold italics*** and concepts from UFO are written in *italics*. In the figure, COM concepts are shown in pink.

Measurable Entity (*Individual*) is anything that can be measured (e.g., a person, a computer system). It is characterized by ***Measurable Elements*** (*Measurable Quality Universals*) which are properties that can be measured. For example, a person can be characterized by his/her weight and height. ***Measure*** is a function (*Quality Function*) used to quantify ***Measurable Elements***, allowing them to be associated with ***Scale Values***, contained in a given ***Scale***. For example, the measure number of lines of source code can be used to quantify the measurable element size of a particular software (***Measurable Entity***). A ***Scale*** (*Measurement Reference Structure*) is a structure that identifies the possible values to which a measure can be mapped (e.g., the set of integer and positive values compose the scale of the measure number of lines of source code). Each value or region that forms a scale is a ***Scale Value*** (*Measurement Reference Region*). ***Measures*** can be expressed in ***Measure Units***, which are units defined and adopted by convention in order to partition ***Scale*** into equal regions (e.g., height can be measured in centimeters). ***Measurement*** is the act of applying a Measure to quantify a ***Measurable Element*** of a ***Measurable Entity***, resulting in a ***Measured Value***. For example, the measurement of the number of lines of source code of a particular software can result in the ***Measured Value*** 34.678.

2.3.1.4 Reference Software Requirements Ontology - RSRO

The SEON domain-specific Reference Software Requirements Ontology (Ruy et al., 2016) is a domain reference ontology that addresses software requirements notions. It is centered in the conception of requirement as a goal to be achieved, and addresses the distinction between functional and non-functional requirements, how requirements are documented in proper artifacts, among others. RSRO is grounded in UFO and reuses SPO and SysSwO. In this work, we address user requirements and how they are related to a user-centred evaluation. Thus, we reuse the general conceptualization about requirements to define relevant HCI concepts (e.g., through specializations). Figure 2.8 presents the RSRO fragment that was reused in this research. In the text, concepts from RSRO are written in underline, concepts from SPO are written in ***bold italics*** and concepts from UFO are written in *italics*. In the figure, RSRO concepts are shown in purple.

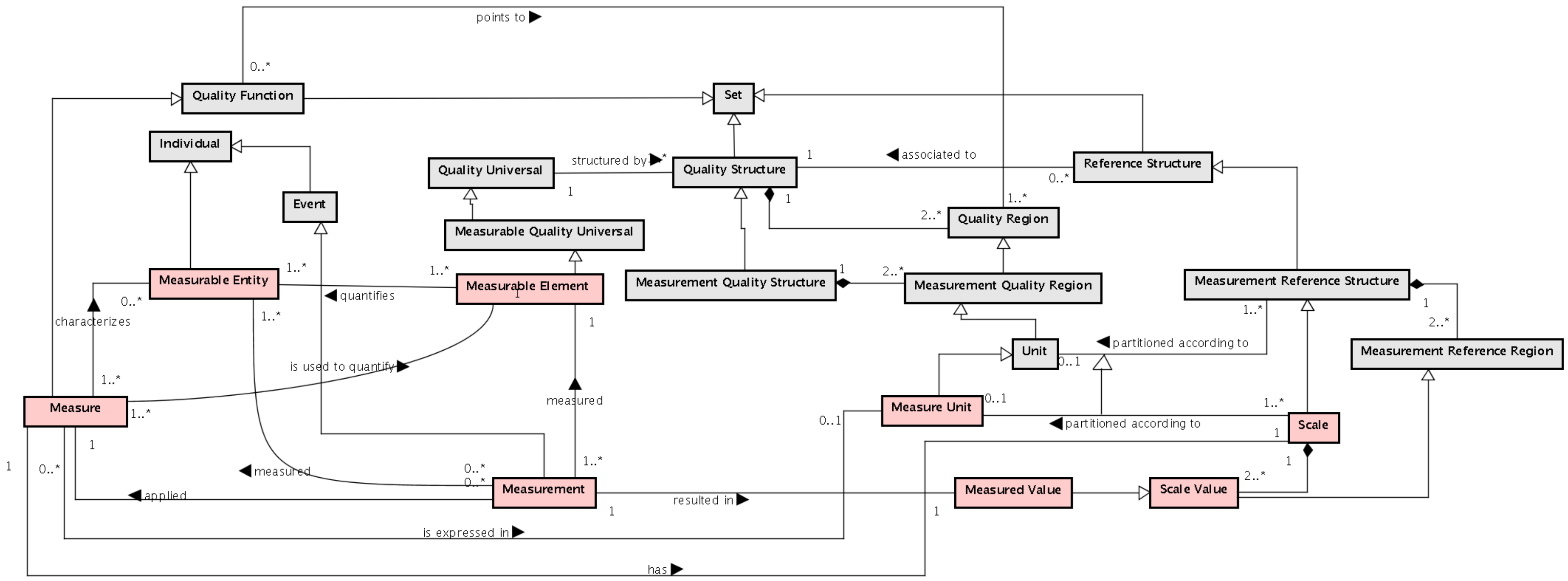


Figure 2.9 – COM fragment relevant for this research.

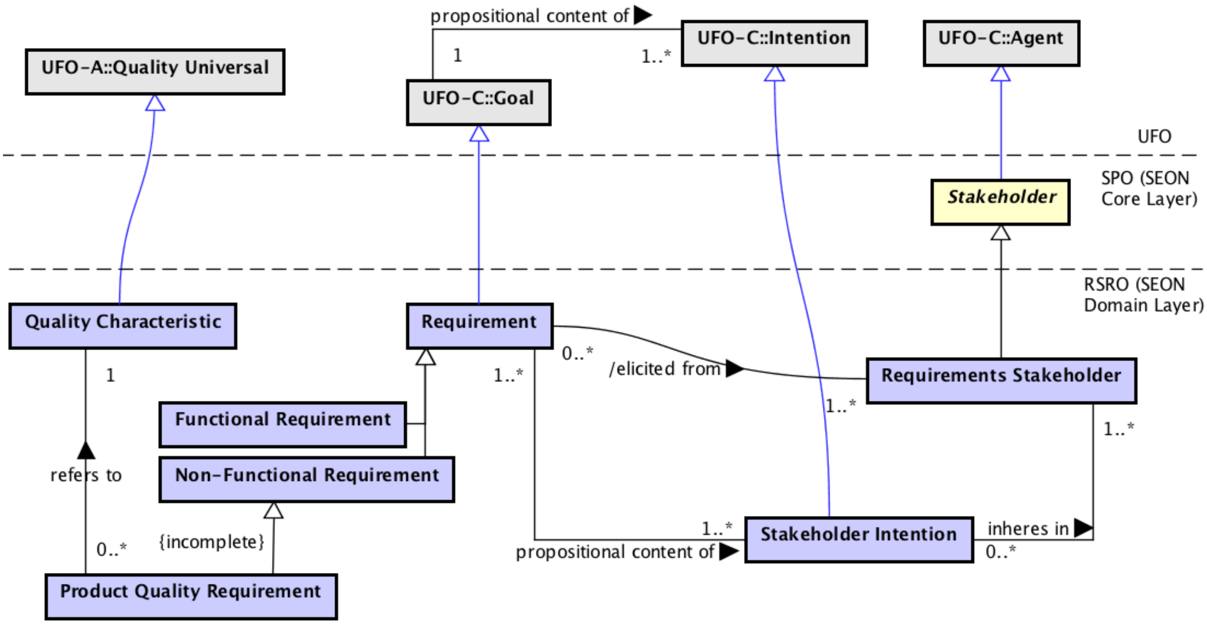


Figure 2.8 – RSRO fragment relevant for this research.

The main RSRO concept is Requirement. It is a *Goal*, representing the users' needs and expectations (Stakeholder Intention) to be achieved as result of the system development. Requirements can be functional or non-functional. Functional Requirements are the ones defining a function to be available from the target system (e.g., the need for providing a client register or controlling an order). They refer to Software Function Types, i.e., types of functions that the software must provide (e.g., providing a client register, controlling an order). Non-Functional Requirements define criteria or capabilities for the system (e.g., being easy to operate, being in conformance with a standard). A special type of Non-Function Requirement is Product Quality Requirement, which refers to Quality Characteristics that the product shall present in some degree, such as reliability, usability, efficiency. The Requirements Stakeholder represents the ***Stakeholders*** from whom the Requirements are collected.

2.4 Final Considerations of the Chapter

This chapter presented the background for this research. It discussed aspects related to HCI, ontologies and ontology networks. Concerning HCI, the phenomenon of communication between user and interactive system, different paradigms were presented, involving several elements that change the way of seeing the phenomenon. It was highlighted that the design and evaluation of interactive systems involve highly specialized knowledge, focused on the user and aiming at usability and user experience. Basic notions of ontologies and ontology networks were introduced and fragments of the ontologies used in this work (i.e., UFO and ontologies from SEON) were also presented. In the next chapter, we present a study in which we investigated HCI ontologies recorded in the literature and their use in the HCI domain.

3 A Secondary study on HCI Ontologies

This chapter presents a secondary study about ontologies in the Human-Computer Interaction domain. The study was performed to help us investigate and understand the research problem. Thus, it is is related to the Relevance cycle (Figure 1.1). The study is also related to the Rigor cycle because it adds knowledge generated by this research to contribute to the knowledge base growth. The study was published in (Costa; Barcellos; Falbo, 2021). Section 3.1 presents the context and motivation for the study. Section 3.2 presents the followed research protocol. Section 3.3 addresses data extraction and synthesis. Section 3.4 discusses the findings. Section 3.5 regards the study limitations. Last, Section 3.6 closes the chapter.

3.1 Context

Considering the HCI knowledge diversity and the successful use of ontologies to solve knowledge-related problems in several domains, we decided to investigate the use of ontologies in the HCI context. To achieve our goal, we performed a secondary study that combines features of systematic literature review (SLR) and systematic mapping (SM). A SLR aims to evaluate and interpret available research studies relative to a topic, area or phenomenon of interest (Kitchenham; Charters, 2007). Indeed, we want to use a trustworthy and auditable method to make a characterization review about ontologies in HCI. However, we also want to provide an overview of this research topic by means of classification and counting contributions in relation to categories of classification schemas, which is consistent with systematic mappings (Kitchenham; Charters, 2007; Petersen; Vakkalanka; Kuzniarz, 2015). For simplification reasons, in the text we refer to our study as an SLR.

Results of a review study help identify gaps in order to suggest future research and provide a direction to appropriately position new research activities (Kitchenham; Charters, 2007; Kitchenham; Budgen; Brereton, 2011; Petersen et al., 2008). In the context of this work, the secondary study aimed at providing a broader view about HCI ontologies, helping us understand the research topic advances thus far and better establishing the scenario for our research.

In this SLR, our objective was to find out (i) *how ontologies have been used to support HCI* and (ii) *how they have been developed*. With (i), we intended to understand which aspects of the HCI domain have been supported by ontologies and the types of ontology-based solutions that have been adopted to solve HCI problems. With (ii), we aimed to verify the quality of the ontologies used in the HCI domain. This is important because the quality of an ontology directly influences the quality of the solution that uses it. Therefore, we investigated how ontologies have been developed, evaluated and the quality characteristics they exhibit.

3.2 Research Protocol

To perform our study, we followed the process defined by (Kitchenham; Charters, 2007), which comprises: ***planning***, when the research protocol is defined with the purpose of supporting study replicability as well as helping researchers to avoid bias when conducting the study; ***conducting***, when the protocol is executed and data are extracted, analyzed and recorded; and ***reporting***, when the results are recorded and made available to potential interested parties.

The ***study goal*** was to understand how ontologies have been used in the HCI domain and how they have been developed. In our study, we refer to these ontologies as HCI ontologies. For achieving this goal, we defined the ***research questions*** presented in Table 3.1.

Table 3.1 – Research questions and their rationale.

RQ	**Research Question**	**Rationale**
RQ1	When and in which type of vehicle have the studies been published?	Provide an understanding on when and where (journal/conference/workshop) publications addressing HCI ontologies have been published, to analyze the maturity of the research topic.
RQ2	Which aspects of the HCI domain have been covered by the existing ontologies?	Identify the aspects of the HCI domain covered by the ontologies to analyze the comprehensiveness of the HCI ontologies and of their use in the HCI domain.
RQ3	What have been the uses of HCI ontologies?	Identify the types of solutions that have been supported by ontologies and understand the purposes of using ontologies in the HCI domain.
RQ4	Have the HCI ontologies been developed by following ontology engineering methods?	Understand how HCI ontologies have been developed, aiming to analyze if good ontology engineering practices have been adopted.
RQ5	How have the HCI ontologies been evaluated?	Verify if (and how) HCI ontologies have been evaluated to analyze if ontology evaluation has been a concern in HCI ontologies.
RQ6	Which quality characteristics have HCI ontologies exhibited?	Identify the HCI ontologies quality characteristics to verify if the ontologies used in the HCI domain present good quality level.

The ***search string*** adopted in the study contains two groups of terms joined with the operator AND, and one group joined with the operator OR. The first group includes terms related to HCI. The second group includes terms related to Ontology. The third group includes refinement to the others. Within the groups, we used OR operator to allow synonyms. The search string is presented in Table 3.2.

Table 3.2 – Search Terms of the SLR.

Search subjects (groups)	Search terms
Area (HCI)	"human-computer interaction" OR "HCI" OR "user interface design" OR "user interaction design" OR "user centered design" OR "user centred design" OR "human-centered design" OR "human-centred design" OR "UI design" OR "user interface evaluation" OR "user interface assessment" OR "user interaction evaluation" OR "user interaction assessment" OR "UI evaluation" OR "UI assessment" OR "user interaction"
Approach (Ontology)	"ontologies" OR "ontology"
Area & Approach (Refinement)	"user interface ontology" OR "user interface ontologies" OR "UI ontology" OR "UI ontologies"
Search String: (("human-computer interaction" OR "HCI" OR "user interface design" OR "user interaction design" OR "user centered design" OR "user centred design" OR "human-centered design" OR "human-centred design" OR "UI design" OR "user interface evaluation" OR "user interface assessment" OR "user interaction evaluation" OR "user interaction assessment" OR "UI evaluation" OR "UI assessment" OR "user interaction") **AND** ("ontologies" OR "ontology")) **OR** ("user interface ontology" OR "user interface ontologies" OR "UI ontology" OR "UI ontologies")	

For defining the search string, we followed the strategy of iteratively improving the search string until obtaining satisfactory results in terms of the number of publications and their relevance (Petersen; Vakkalanka; Kuzniarz, 2015). We started with a very general search string including terms such as "human-computer interaction", "HCI", "user interface" and "ontology". After running some tests, we noticed that the term "user interface" caused many irrelevant papers to be returned. Thus, we refined the search string by combining the term "user interface" (and synonyms) with other terms such as "ontology", "design", "evaluation" and "assessment".

The ***sources*** used in the study were digital libraries that satisfy the following criteria (Costa; Murta, 2013): (i) it allows using logical expressions or a similar mechanism for performing searches; (ii) its search mechanism supports full-length searches; (iii) it is available in the researchers' institution; and (iv) it covers the research area of interest, namely Computer Science.

The following sources were selected: Scopus[1] and Engineering Village[2]. Scopus is one of the largest databases of peer-reviewed literature. It indexes papers from other important sources such as IEEE, ACM and Science Direct, providing useful tools to search, analyze and manage scientific research. Engineering Village is also a citation indexing platform, which indexes complementary sources. Initially, we have also selected the Web of Science database. However, it did not return any additional paper when we searched publications until 2018. Thus, we decided to exclude it from the set of sources.

Publications selection was performed in four steps, as presented in Figure 3.1.

1 <http://www.scopus.com>
2 <http://www.engineeringvillage2.org>

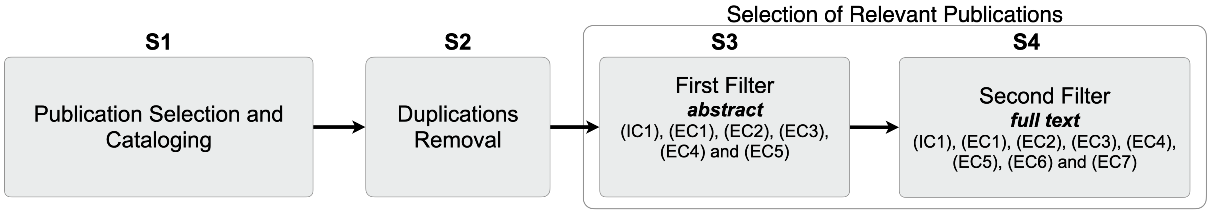


Figure 3.1 – Publication Selection Process.

In *Preliminary Selection and Cataloging (**S1**)*, the search string was applied in the search mechanism of each digital library used as source of publications (we limited the search scope to title, abstract and keywords metadata fields). After that, we performed the *Duplications Removal (**S2**)*, i.e., publications indexed in more than one digital library were identified and duplications were removed. In *Selection of Relevant Publications – 1st filter (**S3**)*, the abstracts of the selected publications were analyzed considering the following inclusion (IC) and exclusion (EC) criteria: (IC1) the paper presents an ontology about the HCI domain; (EC1) the paper does not present an ontology about the HCI domain; (EC2) the paper does not have an abstract; (EC3) the paper was published only as an abstract; (EC4) the paper is a secondary study, tertiary study, editorial, summary of keynote, tutorial or the proceedings of a scientific event; (EC5) the paper is not written in English. Finally, in *Selection of Relevant Publications – 2nd filter (**S4**)*, the full text of the publications selected in S3 were read and analyzed considering the aforementioned criteria plus the following: (EC6) the paper is an older version (outdated) of another paper already considered; and (EC7) we did not have access to the full text of the paper. The (EC6) served to avoid repetition and (EC7) to treat publications whose full text was not available for the researchers.

We used the StArt tool[3] to support publications selection. To consolidate data, publications returned in the publication selection steps were cataloged and stored in spreadsheets. We defined an id for each publication and recorded the publication title, authors, year, and vehicle of publication. Data from publications returned in S4 were extracted and organized into a data extraction table oriented to the research questions. The spreadsheets produced during the study can be found in (Costa; Barcellos; Falbo, 2020c).

Before performing the SLR, we tested the protocol. This test was conducted to verify its feasibility and adequacy, based on a pre-selected set of studies considered relevant to our investigation. The doctorate candidate performed publications selection and data extraction. The advisors reviewed both. Once data has been validated, the doctorate candidate carried out data interpretation and analysis, and again advisors reviewed the results. Disagreements were discussed and resolved. Quantitative data were tabulated and used in graphs and statistical analysis. Finally, the doctorate candidate and advisors performed qualitative analysis considering the findings, their relation to the research questions and the study purpose.

[3] <http://lapes.dc.ufscar.br/tools/start_tool>

3.3 Data Extraction and Synthesis

The study considered papers published until 2019. Searches were conducted for the last time in January 2020. Figure 3.2 illustrates the followed process and the number of publications selected in each step. To search the digital libraries, the search string went through syntactic adaptations according to particularities of each source.

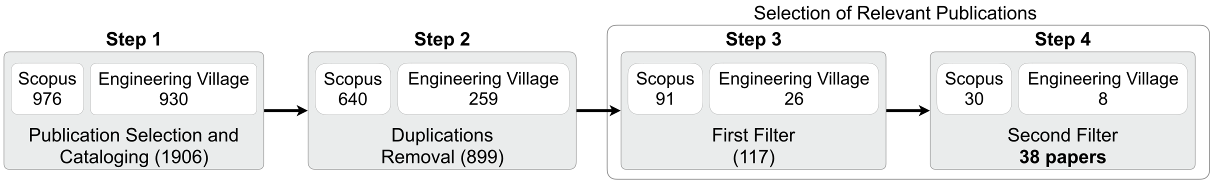


Figure 3.2 – Search and Selection Process.

In the 1st step, as a result of searching the selected sources, a total of 1906 publications was returned. In the 2nd step, we eliminated duplicates, achieving 899 publications (reduction of approximately 53%). In the 3rd step, we applied the selection criteria over the abstract, resulting in 117 papers (reduction of approximately 94%). At this step, we only excluded publications that were clearly unrelated to the subject of interest. In case of doubt, the paper was taken to the next step. Finally, in the 4th step, the selection criteria were applied considering the full text, resulting in 38 publications (reduction of approximately 68%).

From the 38 selected publications, we identified 35 different ontologies because some publications present complementary parts of the same ontology. Table 3.3 shows the list of identified ontologies and corresponding publications. Along this and the next sections we refer to the ontologies by using the id listed in the table. Detailed information about the selected publications, including a brief description and extracted data, can be found in (Costa; Barcellos; Falbo, 2020c). In the following, we present the data synthesis for each research question.

Table 3.3 – Selected Publications and Ontologies.

Id	Ontology	Brief description and some concepts	Reference
#01	Ontology of Multiparty Interaction	It is a conceptual model, based on references such as (Norman, 1986), which addresses HCI phenomenon and the involved parts (i.e., user and system). Includes concepts such as Participants, User and Interaction.	(Storrs, 1994)
#02	UI Design Process Ontology	It is a conceptual model on task modeling and interaction with focus on UI design. Some concepts: Task, Interaction, Action.	(Suàrez; Jùnior; Barros, 2004)
#03	Impairment-User Interface Ontology	It is a conceptual model on UI and adaptation for user impairment. Addresses concepts such as Impairment, User, InterfaceAdaptation.	(Karim; Tjoa, 2006)
#04	DREAMS Ontology	It presents a conceptual model and a small extract of the operational ontology implemented in OWL and RDF that addresses user stereotypes and related aspects for interaction and devices capabilities in massive networking scenarios. Includes concepts such as Social_Role, Device, Input_modality.	(Korfiatis; Constantiou, 2006)
#05	SOUPA Extension	It is a conceptual model which addresses multimodal pervasive systems, devices and environments in multimodal interaction. Some concepts: I/O Device, Sensor, Service.	(Coronato; DE PIETRO, 2007)

Table 3.3 – *Continued from previous page*

Id	Ontology	Brief description and some concepts	Reference
#06	HCI Frame-based Ontology	It is a conceptual model on task modeling and interaction with focus on UI design regarding guidelines structures. Includes concepts such as Target User, Design Property, Usability Requirement.	(Bakaev; Avdeenko, 2010; Bakaev; Avdeenko, 2012)
#07	Ontology for Interactive Adaptive Systems	It is an operational ontology, implemented in OWL, for adaptive interactive systems and its facets such as user, system, user-system interaction and adaptation. Includes concepts such as DialogSystem, Interaction, Adaptation, UserModelItem.	(Bezold; Minker, 2010)
#08	Interaction Ontology	It presents a small extract of an operational ontology implemented in OWL that addresses interaction patterns and web UI regarding structural and visual aspects. Some concepts: Widget, Event, Presentation Properties.	(Celino; Corcoglioniti, 2010)
#09	Abstract Widget Ontology	It is a conceptual model, addressing web UI elements and concepts such as Abstract Interface Element, Element Exhibitor, Variable Capture.	(Martín et al., 2010)
#10	User Interface Ontology	It is an operational ontology implemented using OWL/RDF. It addresses interface entities and relationships, including concepts such as Widget, Textual, Frame.	(Shahzad; Granitzer; Helic, 2011)
#11	Core HCI Ontology	It is a conceptual model of UI design addressing input and output modalities to be chosen according the context of use. Some concepts: Location, Modality, Device.	(Tourwé et al., 2011)
#12	The name is not cited in the paper	It presents a conceptual model and an operational version that addresses interactive modal devices and situations of use. Includes concepts such as Device, DeviceRestrictedByLocation, Situation.	(Bowen; Hinze, 2012)
#13	Gesture Ontology	It is a conceptual model that addresses gesture interaction and aspects involved during the process of producing a gesture. Includes concepts such as User, Body Movements, Function-oriented Gesture.	(Chera; Tsai; Vatavu, 2012)
#14	User Interaction Context Ontology	It is a conceptual model about user interaction and interaction context addressing the characterization of user actions and resources used during his/her interaction. It has an operational version implemented using OWL and includes concepts such as Action, Resource, Task, User.	(Devaurs; Rath; Lindstaedt, 2012)
#15	RaUL Ontology	It is a conceptual model, a markup ontology, about web UI elements related to the structure of a web form. It has an operational version implemented using RDF and includes concepts such as Page, Widget Element, Button.	(Haller; Groza; Rosenberg, 2012)
#16	PersonasOnto	It is a conceptual model about persona characterization and personas aspects related to usability test. It has an operational version implemented using OWL. Some concepts: Person, Persona, Usability Test.	(Negru; Buraga, 2012; Negru; Buraga, 2013)
#17	An ontology for Personas	It is a conceptual model about persona characterization. A small extract of an operational ontology implemented in OWL/RDF is presented. Includes concepts such as Person, Persona, Context, Goal.	(Salma; Eddine; Sabin, 2012; Salma; Marouane, 2016)
#18	Haptic Applications Software Modeling Ontology	It is an operational ontology, implemented using OWL/SWRL about haptic HCI and pathways that haptic information follows between the human and the machine haptic system. Some concepts: Haptic-Devices, Computer haptics, Haptic-Feedback.	(Myrgioti; Bassiliades; Miliou, 2013)
#19	End-to-End Interaction Ontology	It is a conceptual model and an operational ontology (implemented in OWL) that addresses elements of interaction under the WIMP (Windows, Icons, Menus, Pointer) UI perspective. Includes concepts such as Interaction, Interface, User.	(Zamzami; Budiardjo; Suhartanto, 2013)

Table 3.3 – *Continued from previous page*

Id	Ontology	Brief description and some concepts	Reference
#20	Ergonomics of Human-Computer Interface Ontology	It is a conceptual model concerning ergonomics of UI and aspects such as recommendations, graphic design and evaluation. Includes concepts such as HCI Ergonomics, Interface_Evaluation, Design_-And_Evaluation.	(Elyusufi; Seghiouer; Alimam, 2014)
#21	Interaction Event Ontology	It is a conceptual model about interaction event in social media, covering aspects related to users' (social) interactions. Some concepts: Event, InteractionEvent, SocialRole.	(Kabir et al., 2015)
#22	User's Feedback Ontology	It is a conceptual model that addresses user interaction feedback concerning aspects related to structure, credibility and information value. Includes concepts such as Role, Significance, Modality.	(Mezhoudi; Vanderdonckt, 2015)
#23	Web Design Domain Ontology	It is a conceptual model about web UI design property and web UI structure and elements. Some concepts: Interface element, menu, design property, Interface design property.	(Bakaev; Gaedke, 2016)
#24	Ontological Framework for Universal Acess to ICT	It is a conceptual model about universal accessibility in information and communication technologies according to the user needs and preferences. Includes concepts such as UserInterface, UserCharacteristic, UserCondition.	(Koutkias et al., 2016)
#25	Interaction Ontology	It is a conceptual model about craftswoman capacities and appropriated input/output (I/O) interaction modalities, modes and mediums. It is implemented using OWL/SWRL and includes concepts such as Craftswoman, Motor Capacity, Modality, Medium.	(Lebib; Mellah; Mohand-Oussaid, 2016)
#26	Ontological Model of Adaptive User Interface for People with Disabilities	It is a conceptual model about adaptive UI for people with disabilities, covering UI elements and user characterization. An operational version is implemented using Java and Jena. Includes concepts such as User, Disability, Interface Element Type.	(Kultsova et al., 2017)
#27	Usability Guideline Ontology	It is a conceptual model about usability guidelines description and structure. It is implemented using OWL and includes concepts such as Guideline, Guideline Element, Content Type.	(Robal; Marenkov; Kalja, 2017)
#28	PG-Ontology	It is a conceptual model about pervasive game, concerning its description, elements/components and their relationships from the user experience perspective. Includes concepts such as Pervasive_-game, Player, Game.	(Arango-López et al., 2018)
#29	Context Model Centre Ontology	It is a conceptual model about context-awareness, addressing context as the information that is used to characterize the relevant objects and entities that affect computing tasks. Some concepts: Context Domain, Environment Context, User Context.	(Meng, 2018)
#30	Accesibilitic Ontology	It is a conceptual model about characterization of user needs and capabilities, assistive devices/software tools used to support accessibility and the users interaction taking into account capabilities. An operational version is implemented using OWL. Includes concepts such as User, Capacity, ActivityParticipation, SupportAssistance.	(Mariño et al., 2018)
#31	Ontologies of Interaction Channels – Virtual Reality System and Environment	It is a conceptual model about characteristics of interaction channels of Virtual Reality system and environment. Includes concepts such as InteractionChannel, Forms_of_interaction, Spacial_navigation.	(Sokolowski; Walczak, 2018)
#32	Web UI Measurement	It is a conceptual model and an operational ontology (implemented in OWL) about the organization of UI metrics and services used in the UI measurement. Includes concepts such as Service, Metric, Attribute, Interface.	(Bakaev et al., 2019)
#33	Affinto Ontology	It is a conceptual model and an operational ontology (implemented in OWL) about affective HCI, which addresses affective states and different factors of its context of use. Some concepts: Environmental_property, System_property, AffectiveP_properties.	(Garay-Vitoria; Cearreta; Larraza-Mendiluze, 2019)

Table 3.3 – *Continued from previous page*

Id	Ontology	Brief description and some concepts	Reference
#34	IoRT Ontology	It is a conceptual model about the user characteristics and interaction in IoRT (Internet of Robotic Things), addressing relevant aspects for personalized, context-aware task generation. An operational version is implemented using OWL. Includes concepts such as Preferred Interaction, PreferredRobot Interactions, Task, Profile.	(Mahieu et al., 2019)
#35	The name is not cited in the paper	It is a conceptual model and an operational ontology (implemented in OWL) about the presentation and behavior of interactive components (graphical UI) used in web and mobile applications. It also addresses the structure of user stories, tasks, scenarios and prototypes. Some concepts: Event, Action, Condition, Menu, clickOn.	(Silva; Winckler; Trætteberg, 2019)

3.3.1 Publication year and vehicle (RQ1)

Figure 3.3 shows the distribution of the 38 selected publications over the years and the distribution of publications considering the vehicle of publication. Papers addressing ontologies in the HCI context have been published since 1994 in journals and conferences (no workshop publication was found). Conferences have been the main forum, encompassing 68.4% of the publications (26 out of 38). 12 papers (31.6%) were published in journals.

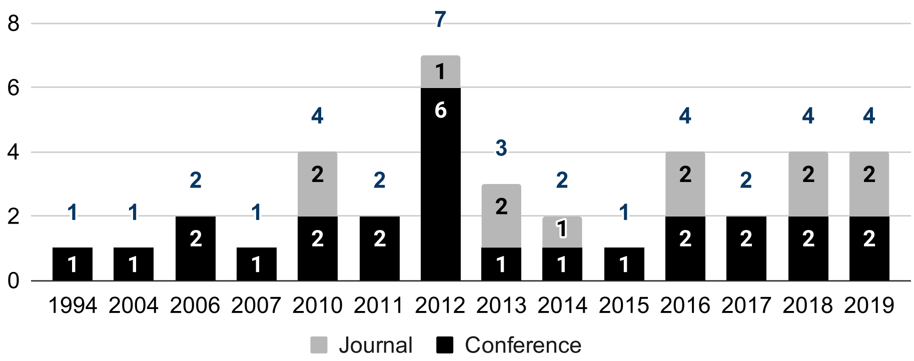


Figure 3.3 – Publications over the years.

Table 3.4 presents the publication sources of the selected studies, their types and #id. Publications vehicles in areas such as Information Technology and Artificial Intelligence seem to be more receptive for presenting studies related to the investigated research topic. Moreover, only two papers (#30 and #33) were published in the same journal (IEE Access) while the others were published in different publication sources.

Table 3.4 – Publication Sources.

Publication source	Type	Id
Interacting with Computers	Journal	#01
Conference on Task models and diagrams (TAMODIA'04)	Conference	#02
International Conference on Computers for Handicapped Persons (ICCHP) Computers Helping People with Special Needs	Conference	#03

Table 3.4 – *Continued from previous page*

Publication source	**Type**	**Id**
European Conference on Information Systems (ECIS'06)	Conference	#04
International Conference on Security and Privacy in Communication Networks (Secure-Comm'07)	Conference	#05
The IASTED International Conference on Automation, Control, and Information Technology - Information and Communication Technology (ACIT-ICT'10)	Conference	#06
International Conference on Database Systems for Advanced Applications (DASFAA'12)	Conference	#06
Journal of Ambient Intelligence and Smart Environments	Journal	#07
International Conference on Semantic Systems	Conference	#08
World Wide Web	Journal	#09
International Conference on Computer Sciences and Convergence Information Technology (ICCIT'11)	Conference	#10
Belgian/Netherlands Artificial Intelligence Conference (BNAIC'11)	Conference	#11
ACM SIGCHI symposium on Engineering interactive computing systems (EICS'12)	Conference	#12
IEEE International Symposium on Intelligent Control	Conference	#13
Applied Artificial Intelligence	Journal	#14
Joint International Semantic Technology Conference (JIST'11)	Conference	#15
International Conference on Knowledge Engineering and Ontology Development (KEOD'12)	Conference	#16
International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management (IC3K'12)	Conference	#16
IEEE International Conference on Complex Systems (ICCS'12)	Conference	#17
Journal of Theoretical and Applied Information Technology	Journal	#17
Expert Systems with Applications	Journal	#18
International Journal of Software Engineering and its Applications	Journal	#19
International Conference on Multimedia Computing and Systems (ICMCS'14)	Conference	#20
The Computer Journal	Journal	#21
World Symposium on Web Applications and Networking (WSWAN'15)	Conference	#22
IEEE North West Russia Section Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW'16)	Conference	#23
Universal Access in the Information Society	Journal	#24
International Conference on Web Information Systems and Technologies (WEBIST'16)	Conference	#25
Conference on Creativity in Intelligent Technologies and Data Science (CIT&DS'17)	Conference	#26
Portland International Conference on Management of Engineering and Technology (PICMET'17)	Conference	#27
World Conference on Information Systems and Technologies (WorldCIST'18)	Conference	#28
KSII Transactions on Internet and Information Systems	Journal	#29
IEEE Access	Journal	#30, #33
Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS'18)	Conference	#31
International Conference on Web Engineering (ICWE'19)	Conference	#32
Journal of Systems and Software	Journal	#34
IFIP Conference on Human-Computer Interaction (INTERACT'19)	Conference	#35

3.3.2 Coverage of the HCI domain (RQ2)

Figure 3.4 shows the HCI aspects addressed by the ontologies identified in the study. The categorization used to classify the aspects was established based on the data extracted. First, we extracted information as recorded in the publications (e.g., ontology #13 addresses the HCI phenomenon by means of gesture). Then, we analyzed data extracted and identified categories to group them. For example, we considered in the "HCI phenomenon" category all ontologies addressing the HCI phenomenon, regardless the mode of interaction.

User Interface (UI) has been the HCI aspect most addressed by the HCI ontologies, being covered by 12 of them (34.29%). In this context, some ontologies focus on the UI elements (#09, #10, #15, #19, #23, #26, #35), addressing them from graphical (#10, #35), web (#09, #15, #23, #26) and WIMP (#19) perspectives. Other ontologies address other aspects of UI, such as UI adaption (#03) and UI measurement (#34). *HCI phenomenon* has been the second HCI aspect most addressed, being treated in 11 ontologies (31.43%). Some ontologies deal with specific modes of interactions such as gesture (#13), haptic (#18), virtual reality (#31), affective (#33) and IoRT (#34). #25, in turn, addresses several modes of interaction, representing them in categories such as direct manipulation, language, visual, hearing and tactile. #01 describes HCI as multiparty, concerning one or more participants. #14, #19, #21 and #35 represent HCI by means of events, actions and tasks. *User characterization* and *HCI or UI Design* have been, respectively, the third and fourth most addressed HCI aspect. The former is treated in 10 ontologies (28.57%) that address user impairment (#03), social roles (#04), user modeling (#07), persona (#16, #17), user capacity (#25, #30), disability (#26), needs and preferences (#24, #34). Nine ontologies (25.71%) concern the latter, being #13 focused on HCI design and #02, #06, #08, #11, #16, #17, #20, #23 on UI design.

Besides the four aforementioned aspects, HCI ontologies have also addressed other nine HCI aspects. *Context of use/Context-awareness* is treated in six ontologies (17.14%) that address interaction context (#14), context-awareness in mobile adaptation (#29), context-based approach in virtual reality (#31), context of use factors that influence affective states (#33) context-aware systems for personalized interaction tasks in IoRT, Four of these ontologies (73.33%) also deal with the phenomenon of interaction.

Five (14.29%) ontologies address *Accessibility*, dealing with UI accessibility (#03, #25, #26) as well as with the user needs (#24, #30). Four (11.43%) ontologies treat *Multimodal*

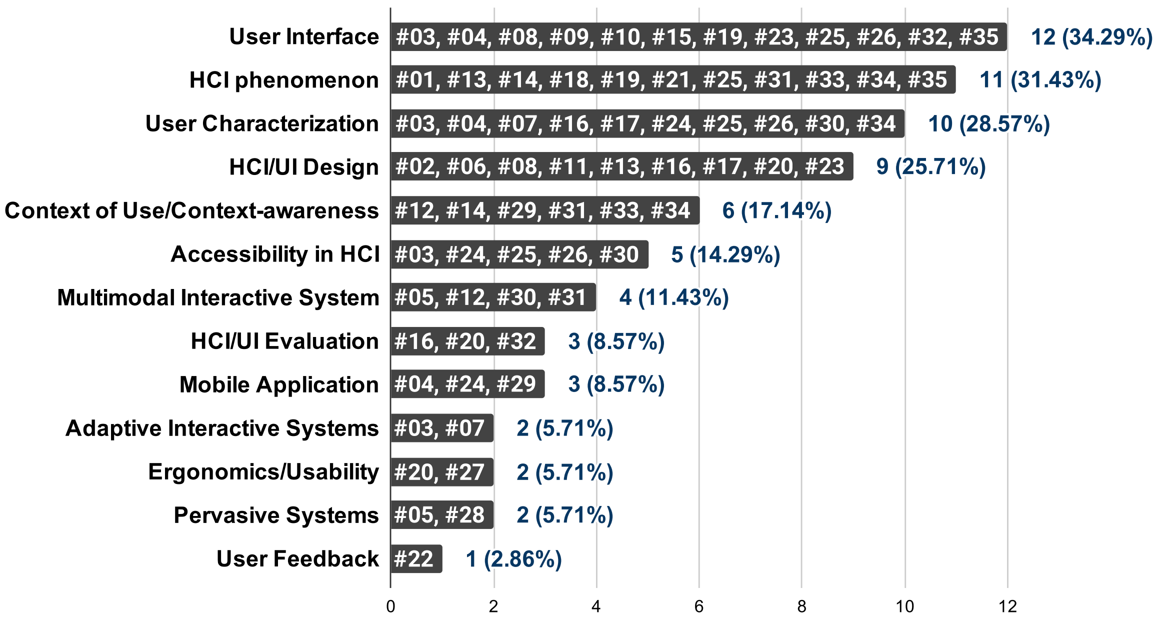


Figure 3.4 – HCI aspects covered by the found ontologies.

Interactive Systems, addressing multimodal devices (#12, #30) and multimodal devices and systems (#05, #31). *HCI/UI Evaluation* is treated in three (8.57%) ontologies. #16 encompasses usability test under the persona perspective, #20 concerns UI ergonomics and #32 regards UI measurement. Also three (8.57%) ontologies address *Mobile Application*, dealing with mobile devices capacities (#04), accessibility (#24) and performance (#29).

The four less addressed aspects were treated only in two (5.71%) or one (2.86%) ontology. Ontologies addressing *Adaptive Interactive Systems* have dealing with adaptive interactive systems as a whole (#07) and UI adaptation for user impairment (#03). In the *Ergonomics/Usability* context, #20 addresses ergonomics by means of recommendations, while #27 describes usability guidelines. In its turn, *Pervasive Systems* has encompassed ontologies addressing pervasive game description (#28) and proactive environments (#05). Finally, #22 treats *User Feedback* description concerning its structure, credibility and information value.

3.3.3 Use of Ontologies in HCI (RQ3)

When investigating how ontologies have been used in the HCI domain, we first looked at the types of ontology-based solutions that have been used and, thus, we investigated their use in the HCI domain. Figure 3.5 shows the identified types of ontology-based solutions.

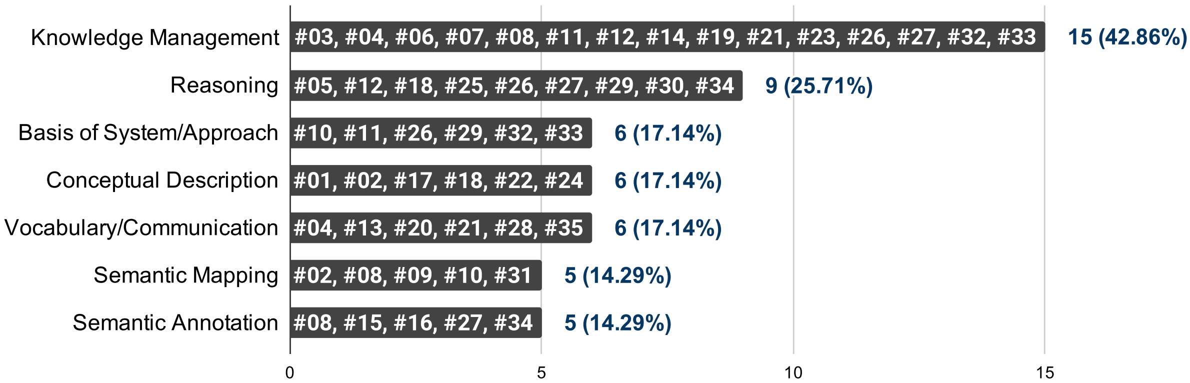


Figure 3.5 – Ontology-based solutions.

The most common ontology-based solution has been *Knowledge Management*, which has been the use of 15 out of 35 ontologies (42.86%). In this category, ontologies have been used to support knowledge capture, representation, organization, use, sharing, among others. In this context, #06, #07, #12, #14, #23, #26, #27 and #33 were used to structure knowledge bases. For example, #06 was used in a recommender system supporting HCI design for web applications; #07 served as a semantic representation to an interactive system, enabling the UI to adapt itself to each user; #23 structured knowledge in an Evolutionary Algorithm approach, which combines Artificial Intelligence and design methods to automate development and improvement of web UI; #27 was used in a tool to support automatic evaluation of web UIs based on usability guidelines; and #33 was used as a basis of an affective multimodal conversational system. Some ontologies were used to knowledge capture and sharing. For example, #11 was used in an ontology-driven modeling framework to capture and manage design expert knowledge to be provided as design-related recommendations. #03 was used in a user's personal information management system to capture knowledge and adapt the UI according to the user impairments. By using a template to capture knowledge, #19 supported reverse engineering requirements for systems without documentation or available source code. #14 was used to increase the performance of user task detection by capturing simple interaction events (key pressed and mouse clicks) from the user's computer desktop which are automatically stored into a repository for further inferences. The other three ontologies (#04, #21, #32) were used to knowledge representation. #04 regards the representation of social context and its surrounding activities to support the development of mobile service provision models, #21 represents knowledge about the user's interaction events in different social media and #32 structures knowledge about web UI metrics.

Nine (25.71%) ontologies were used to *Reasoning*, i.e., to derive information that is not explicitly expressed in the ontology or in the knowledge base (Victoria; Antoanela; Cicortas, 2008). For example, #25 was used to determine the most suitable and adaptable user interfaces to craftswoman (user) considering modalities and sensory and motor abilities; #34 was used to build services that generate interaction tasks for social robots in smart IoT environments; and #12 was used to reason about the usability of interactive medical devices in multiple situations of use.

The third main use of HCI ontologies, with six out of 35 (17.14%) ontologies, has been *Basis of a System/Approach*, *Conceptual Description* and *Vocabulary/Communication*. Being a *Basis of a System/Approach* means that the ontology is used as part (e.g., a module or a repository) of the proposed system/approach architecture. This is the case, for example, of #29 and #34. The former is part of a context-aware computing architecture, supporting architecture layers such as knowledge discovery and context-aware services. The latter is part of the reasoning service component of a semantic-based platform architecture conceived as a middleware.

Ontologies used for *Conceptual Description* aim to provide a formal specification of a portion of the domain of interest. In this context, #18 provides a conceptualization regarding haptic interaction; #01 describes the HCI phenomenon, #2 conceptualizes interaction based on task description; #17 provides a conceptual representation about personas; #22 presents a conceptual description about user feedback; and #24 provides a semantic representation of concepts that are required for addressing universal accessibility related to user interaction with Information and Communication Technologies.

Ontologies used for *Vocabulary/Communication*, as the name suggests, are used as a vocabulary or for communication purposes. In this sense, #35 was used as a common vocabulary in an approach to allow for automatic assessment of the graphical UI prototypes against user stories. #04 supported communication related to social context and its surrounding activities to aid the development of advanced models of mobile service. #21 represents user interaction events in different social media and it was used as a vocabulary for semantic event matching. #13 helped to inform the design of service-oriented architectures for engineering new systems and applications. #20 was used to support to build user profiles. Finally, #28 served as a consensual instrument to talk about pervasive games from the user experience perspective.

Five out of 35 (14.29%) ontologies were used to *Semantic Mapping*, i.e., interrelate concepts from different sources (e.g., data bases, documents, ontologies, etc.). For instance, #31 was used to map concepts related to virtual reality (VR) system and VR environment, enabling the proposed system to assign interaction channels to the interaction capabilities of the VR environment. Finally, five (14.29%) ontologies were used to *Semantic Annotation*, meaning that semantic information was added to artifacts, allowing machines to read them. For example, in #27, each usability guideline was annotated enabling information to be accessed in a machine-processable form to automatically access the UI elements against selected UI guidelines and the associated metrics.

Table 3.5 summarizes the aspects addressed by each HCI ontology (RQ2) and the ontology-based solutions on which the ontologies were used (RQ3).

3.3.4 Ontology Development (RQ4)

According to the publications, the development of only three (8.57%) out of 35 ontologies followed an ontology engineering method (see Figure 3.6): #25 adopted Methontology (Fernández-López; Gómez-Pérez; Juristo, 1997), #28 used Methontology and NeOn (Gómez-Pérez; Suárez-Figueroa, 2009; Suárez-Figueroa et al., 2012) and #30 applied NeOn. Although other publications (#02, #11, #12, #13 and #18) mention the use of some guidelines to develop the ontologies, the methods are not clearly presented or referred. Thus, we did not consider that they adopted an ontology engineering method.

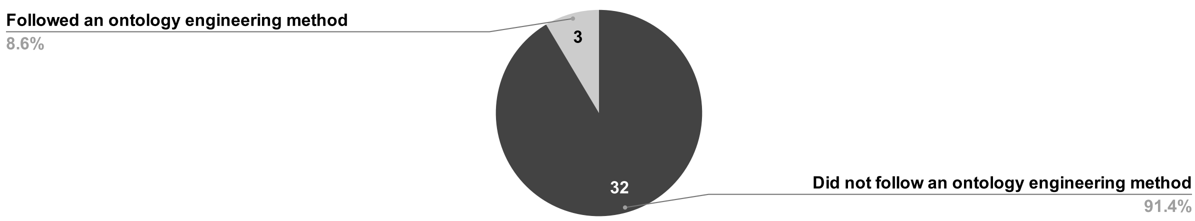


Figure 3.6 – Ontology development.

3.3.5 Ontology Evaluation (RQ5)

In this research question, we used the classification proposed by Brank, Grobelnik and Mladenić (2005), namely: task-based evaluation and data-driven evaluation. In a task-based evaluation, the ontology is used in some kind of application or task, and the outputs of the application, or its performance on the given task, are used to evaluate the ontology (Brank; Grobelnik; Mladenić, 2005). This is an indirect way of evaluating an ontology and, depending on the complexity or

Table 3.5 – HCI ontologies: HCI aspects and ontology-based solutions.

	HCI covered aspects													**Ontology-based solution**						
Id	UI	HCI Phenomenon	User Characterization	HCI/UI Design	Context of Use / Context-awareness	Acessibility in HCI	Multimodal Interactive System	HCI/UI Evaluation	Mobile Application	Adaptive Interactive Systems	Ergonomic /Usability	Pervasive Systems	User Feedback	KM	Reasoning	Basis of a System	Conceptual Description	Vocabulary	Semantic Mapping	Semantic Annotation
#01		√															√			
#02				√													√		√	
#03	√		√			√				√				√						
#04	√		√						√					√				√		
#05							√					√			√					
#06				√										√						
#07			√							√				√						
#08	√			√										√					√	√
#09	√																		√	
#10	√															√			√	
#11				√										√		√				
#12					√		√							√	√					
#13		√		√														√		
#14		√			√									√						
#15	√																			√
#16			√	√				√												√
#17			√	√													√			
#18		√													√		√			
#19	√	√												√						
#20				√				√			√							√		
#21		√												√				√		
#22													√				√			
#23	√			√										√						
#24			√			√			√								√			
#25	√	√	√			√									√					
#26	√		√			√								√	√	√				
#27											√			√	√					√
#28												√						√		
#29					√				√						√	√				
#30			√			√	√								√					
#31		√			√		√												√	
#32	√							√						√		√				
#33		√			√									√		√				
#34			√		√										√					√
#35	√	√																√		

subjectivity of the application, it is hard to say if the ontology is really correct. In a data-driven evaluation, the ontology is compared to existing data about the problem domain to which the ontology refers (Brank; Grobelnik; Mladenić, 2005). As shown in Figure 3.7, 15 (42.86%) ontologies (#02, #05, #06, #07, #08, #10, #12, #14, #15, #16, #21, #26, #29, #31, #35) were evaluated by following a task-based evaluation, two (5.71%%) ontologies (#09, #30) used a data-driven evaluation method, five (14.29%) ontologies (#18, #19, #25, #27, #33) used both methods and 13 (37.14%) ontologies (#01, #03, #04, #11, #13, #17, #20, #22, #23, #24, #28, #32, #34) were not evaluated.

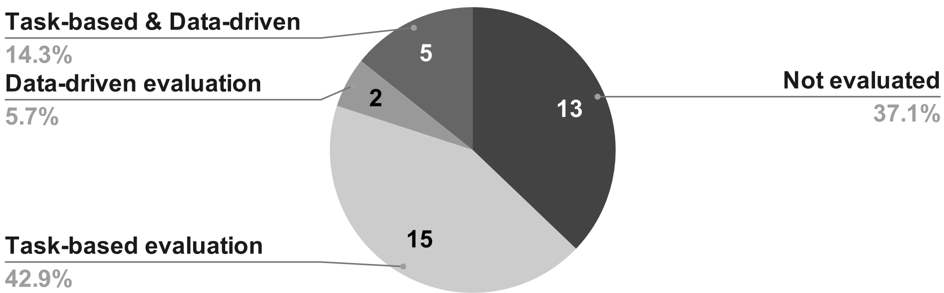


Figure 3.7 – Ontology evaluation.

20 (51.14%) ontologies (#02, #05, #06, #07, #08, #10, #12, #14, #15, #16, #18, #19, #21, #25, #26, #27, #29, #31, #33, #35) were evaluated through a task-based method. For example, the evaluation of #25 and #31 was carried out considering prototypes built using the ontologies. Analogously, systems built using the ontologies were used to evaluate #06 (university website and recommender system), #07 (interactive dashboard), #12 (safety-critical system), #26 (hospital device system) and #27 (interface customization system). #15 was evaluated through a RESTful Web Service that was used to deploy, generate and retrieve RDFa annotated web forms using the ontology. In #16, a semantic HTML template was applied in a small-scale experiment to build a web-based application for managing academic information (e.g., grades, schedule). #19 was evaluated by means of a template used to recover system requirements from the actions performed by the user.

Seven (17.5%) ontologies (#09, #18, #19, #25, #27, #30, #33) adopted a data-driven approach. All of them used some kind of instantiation to evaluate the ontology. For example, #19 was instantiated considering a real case of requirements recovery, while #27 was evaluated through the instantiation of 115 usability guidelines. #30 was the only ontology evaluated by using a data-driven approach and that used competency questions to represent the ontology requirements. In its evaluation, #30 instantiation was oriented to the competency questions.

Finally, 13 (37,14%) ontologies (#01, #03, #04, #11, #13, #17, #20, #22, #23, #24, #28, #32, #34) were not evaluated. In some cases, the authors stated that the ontology was under evaluation (e.g., #24). In others, toy examples or workshops were used to illustrate the ontology use (e.g., #11, #34). Based on the information provided in the publications, we did not consider that these ontologies were evaluated because they were not actually used in real situations nor instantiated real data.

3.3.6 Ontology Quality Characteristics (RQ6)

In this research question, we considered some of the characteristics of "beautiful ontologies" defined by D'Aquin and Gangemi (2011)[4], namely: (i) reusing foundational ontologies; (ii) being formally rigorous; (iii) implementing also non-taxonomic relations; (iv) being modular or embedded in modular framework; (v) implementing an international standard; (vi) being based on competency questions, and (vii) following an evaluation method. Table 3.6 presents the quality characteristics identified in the ontologies.

Table 3.6 – Research questions and their rationale.

id	reusing foundational ontologies	being formally rigorous	implementing also non-taxonomic relations	being modular	implementing an international standard	based on competency questions	following an evaluation method
#01			√				
#02		√	√	√			√
#03			√	√			
#04			√	√	√		
#05		√	√	√			√
#06		√	√				√
#07		√	√	√			√
#08			√	√			√
#09			√				√
#10			√	√			√
#11		√	√	√		√	
#12		√	√			√	√
#13			√	√		√	
#14				√			√
#15				√			√
#16			√	√			√
#17			√				
#18	√	√	√	√			√
#19			√	√			√
#20				√			
#21		√	√				√
#22							
#23				√			
#24		√	√	√	√		
#25		√	√				√
#26		√	√	√			√
#27			√				√
#28			√		√	√	
#29							√
#30		√	√	√		√	√
#31				√			√
#32			√	√	√		

[4] We did not consider characteristics such as commercial impact, which are difficult to evaluate based only on information provided in the publications.

Table 3.6 – *Continued from previous page*

id	reusing foundational ontologies	being formally rigorous	implementing also non-taxonomic relations	being modular	implementing an international standard	based on competency questions	following an evaluation method
#33		√	√	√	√		√
#34	√	√	√	√	√		
#35							√

Concerning the reuse of foundational ontologies, only #18 and #34 (5.71%) were developed based on a foundational ontology (Basic Formal Ontology – BFO[5] (Grenon; Smith; Goldberg, 2004) and Dolce Ultralight upper ontology[6] (Gangemi et al., 2002), respectively).

With respect to being formally rigorous, we considered two aspects: whether axioms/rules are defined, and how the ontology is presented (as a graphical conceptual model, as an operational ontology, using some formalism, or textual description only). Eight (22.86%) ontologies define rules (#02, #05, #06, #07, #11, #24, #25, #26), two (5.71%) ontologies define axioms (#30, #33) and four (11.43%) ontologies define both (#12, #18, #21, #34). Thus, 14 ontologies (40%) present some degree of formalism. Only one (7.14%) of those 14 ontologies (#02) is presented by means of textual description (since it defined rules, we consider that it presents some degree of formalism). Two (14.29%) (#07, #18) are operational ontologies, being represented as OWL and OWL SWRL artifacts, respectively. Five (35.71%) ontologies (#06, #11, #21, #24, #26) are presented as graphical conceptual models. The remaining six (42.86%) ontologies (#05, #12, #25, #30, #33, #34) are presented as both, graphical conceptual model and operational ontology.

Regarding the types of relations, 27 (77.14%) ontologies consider both taxonomic and non-taxonomic relations (although #01, #10 do not clearly represent them). Six (17.14%) ontologies (#14, #15, #20, #23, #29, #31) consider only taxonomic relations. In two (5.71%) ontologies (#22, #35) it was not possible to identify the types of relations.

As for modularity, we investigated if the ontology was organized in hierarchy, dimension, abstraction levels, structure of classes, or if it is non-modular. 12 (34.29%) ontologies are non-modular, while 23 (65.71%) ontologies are modularized according to one of the aforementioned modularization types.

Concerning the use of standards, only four (11.43%) ontologies mention their use as a reference to the ontology conceptualization: #04 used the Composite Capabilities/Preference Profiles part of the Device Independence activity of W3C (Klyne et al., 2004), #24 considered ISO/IEC 24751 (ISO/IEC, 2008a), ETSI TS 202 746 (ES, 2010) and ISO 9999:2011 (ISO, 2011), #32 used W3C Web Content Accessibility Guidelines[7], #33 mentions models found in

5 <http://ontology.buffalo.edu/bfo/>
6 <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>
7 <https://www.w3.org/WAI/ER/tools/>

the literature, and #34 considered the vcard specification (RFC6350[8]) and the W3C Organization Ontology[9]. Other works (#01, #02, #03, #10, #13, #16, #17, #18, #28, #33) mention the use of references but do not specify which ones were used. #28 and #33 mention models found in the literature. Therefore, we did not consider that they use international standards.

Regarding the use of competency questions, only five (14.29%) ontologies (#11, #12, #13, #28 and #30) used competency questions to establish the ontology requirements.

3.4 Data Analysis and Discussion

In this section, we provide additional information about the analyzed HCI ontologies and make some discussion about the obtained results.

Considering *when the studies have been published*, although the first publication identified in the study was published in 1994, HCI ontologies only became a regular research topic addressed in publications from 2010. We believe that this is due to the increasing use of ontologies in several domains in the last decade (ontologies became particularly popular because the Semantic Web) and the acknowledgement of the advantages of using them to solve knowledge-related problems. Regarding the *type of vehicle*, we noticed a predominance of conference papers (26 out 38 papers). The low percentage of journal publications, which generally require more mature works, can be seen as a sign that the research on this topic is not mature enough yet. We also noticed that there is no “home” or well-established forum for publications addressing HCI ontologies, since the 38 papers were published in many different conferences and journals. In fact, it seems that papers addressing HCI ontologies have been more welcome in vehicles of areas such as Information Technology and Artificial Intelligence than in vehicles devoted to HCI.

With respect to *the HCI aspects covered by the ontologies*, the top three aspects have been user interface (12 ontologies), interaction phenomenon (11 ontologies) and user characterization (10 ontologies). These aspects concern core knowledge of the HCI domain. When talking about HCI, it is often necessary to talk about the HCI phenomenon itself and the parts involved in it, i.e., the user and the system, which is perceived by means of the user interface. Thus, it was expected that these aspects were focus of many ontologies. HCI or UI design has also been addressed by many ontologies (nine). This shows that representing knowledge related to designing interactive systems has been considered relevant. Other aspects have not been much explored (e.g., user feedback, usability, HCI or UI evaluation) and others have not even been considered (e.g., user experience, prototype), suggesting that there are still many portions of the HCI domain that can be addressed by ontologies.

Considering *how ontologies have been used in the HCI domain*, there has been a predominance of knowledge management solutions (15 ontologies) and reasoning (9 ontologies).

8 <https://www.w3.org/TR/vcard-rdf/>
9 <https://www.w3.org/TR/vocab-org/>

Ontologies are useful to structure knowledge. Thus, knowledge management has been an ontology use of paramount importance in several domains (Souza; Falbo; Vijaykumar, 2015). The study results showed that the HCI domain has also benefited from the use of ontologies in this kind of application. Ontologies as conceptual models as well as operational ontologies have been used in knowledge management solutions. On the other hand, reasoning has involved the use of operational ontologies to support getting new information from data captured by an application or stored in a repository. By combining results from RQ2 and RQ3, we can notice that the two more used ontology-based solutions (knowledge management and reasoning) have been used only in the context of the four more addressed HCI aspects (UI, HCI phenomenon, user characterization and HCI/UI design). In line with what we said before, we believe that these aspects have been much explored because they are core aspects in the HCI domain. Thus, we think that it would be expected that they were addressed in the ontology-based solutions most used in the HCI domain. Although HCI/UI evaluation is an important aspect of the HCI domain (Preece; Sharp; Rogers, 2015), we noticed that it has not been much explored.

By analyzing the ontologies use reported in the selected publications, we can point out some advantages that have been obtained from the use of ontology-based solutions to solve HCI problems: (i) by using ontologies it has been possible to represent knowledge about the domain in a clear and explicit way, supporting communication and learning tasks. For example, #01 and #28 describe and make clear knowledge about HCI phenomenon and pervasive games, respectively. (ii) ontologies have allowed to structure knowledge, supporting the development of ontology-based systems, frameworks, approaches, architectures, etc. For example, #07 is used (both at design time and at runtime) in a framework for adapting interactive systems according to the user behavior. (iii) ontologies have enabled automated solutions and reasoning. For instance, #12 is used to reason about the usability of interactive medical devices in multiple situations of use. (iv) ontologies have supported semantic interoperability, enabling data/information from different sources be integrated in HCI solutions (e.g., #16 it is used to semantically annotate a template about persona to integrate multiple data).

Concerning the *quality of HCI ontologies*, we noticed that although the ontologies present some quality characteristics (e.g., being modular and implementing non-taxonomic relations), others have not been a concern. For example, only one ontology (#34) is grounded in a foundational ontology. Foundational ontologies provide basic concepts and help domain ontologies to be truthful. Most of the HCI ontologies are not formally rigorous. For example, the ontologies presented in #13, #15, #20, #23, #29 and #31 are, in fact, taxonomies or lightweight ontologies, while #07, #08, #10 and #18 are just OWL artifacts (i.e., operational ontologies). Being formally rigorous makes the ontology more precise. Only five ontologies used international standards as a basis. Standards are important references to ontologies because they provide a consensual conceptualization about the domain of discourse. As for ontology development, only three were developed by following an ontology engineering method. By following an ontology engineering method, one adopts best practices to increase ontology quality. In this context, only three ontologies had their scope defined by means of competency questions. These questions help to elicit the ontology requirements and produce an ontology that meet the user needs. Most HCI ontologies have been evaluated. However, evaluation has not been an explicit subject in the publications. Most of the time we considered that the instantiations or uses made of the ontologies were a way of evaluating them. The quality of an ontology directly affects the solution built using it. Thus, it is of paramount importance to use good practices to develop HCI ontologies. In general, the study results showed that ontology engineering practices and principles has not been a concern when developing HCI ontologies. We think that this can be seen as an opportunity to HCI and Ontology Engineer professionals and researcher get closer.

Finally, by analyzing the ontologies coverage (we extracted the ontologies concepts, as it can be seen at (Costa; Barcellos; Falbo, 2020c), we noticed several overlaps. Although several ontologies address the same concepts or complementary portions of the HCI domain, they do not reuse each other's concepts. In fact, we noticed that HCI ontologies have been developed for solving specific problems, in specific contexts, without a concern with integration to other existing HCI ontologies. This causes an increasing number of ontologies and makes it difficult to integrate different ontologies to produce a new ontology covering a larger portion of the domain. This, in turn, leads one to create new ontologies instead of reusing existing ones, feeding a 'vicious cycle' of many overlapping ontologies.

3.5 Limitations of this Secondary Study

Although the ambition of a secondary study is to summarize all relevant research in an area, different sets of publications can be obtained given a number of decisions and judgment taken (Wohlin et al., 2012). Thus, as any study, the study presented in this chapter has some limitations. In addition, some challenges can reach the researchers during a secondary study, such as how to select a comprehensive and relevant source of publications, how to consistently apply the inclusion/exclusion criteria, how to classify data and how to interpret them. In this study, we experienced these challenges and carried out some actions aiming at minimazing their influence on the results.

Publication selection and data extraction were initially performed by one of the authors. To reduce this subjectivity, the other two authors performed these same steps over a random sample of about 45% of the non-duplicated publications. Thus, at least 400 publications were evaluated by two researchers. The results of each reviewer were then compared. Disagreements and possible biases were discussed in meetings. Concerning the 38 selected publications, all of them were evaluated by at least two researchers.

An analysis of degree of concordance was performed to measure the level of agreement between the results obtained from the researchers in the selection process. For this, we calculated the kappa coefficient (Landis; Koch, 1977). Kappa is a statistical measure of inter-rater agreement for qualitative (categorical) items. It is considered to be a more robust measure than simple percent agreement calculation, since it takes into account the agreement occurring by chance. Kappa maximum value is 1, representing total agreement. Values close to and below 0 indicate no agreement. Considering the set of publications analyzed by all the reviewers, the obtained kappa coefficient was 0.79, which, according to Landis and Koch (1977), means substantial agreement.

Terminological problems in the search strings may have led to missing some primary studies. In order to minimize these problems, we performed previous simulations in the selected databases. We decided not to search any specific conference proceedings, journals, or the grey literature (technical reports and works in progress). Thus, we have just worked with studies indexed by the selected electronic databases. The exclusion of these other sources makes the review more repeatable, but possibly some valuable studies may have been left out of our analysis. Although the digital libraries adopted represents a comprehensive source of publications, the exclusion of other sources and the fact that we did not performed snowballing may have left some valuable publications out of our analysis.

Applying the search string to digital libraries engine can also interfere in the returned publications. First, because it is necessary to adapt the string syntax. The adaptation may result in a different search string. The string used in the study was adapted to each digital library and we double-checked the adapted string and ran tests to ensure that it was correct. Second, because of problems or changes in the engines. During the study, changes occurred in the engine of one of the used digital libraries. We ran some tests that showed us that the engine interface had changed, but the engine seemed to work as before.

Another limitation is related to the classifications we made for categorizing data. Some categories were based on classifications proposed in the literature (e.g., quality characteristics (D'Aquin; Gangemi, 2011), evaluation types (Brank; Grobelnik; Mladenić, 2005)). Others, were established during data extraction, based on data provided by the analyzed publications (e.g., categories used in RQ2 and RQ3). Classification schemas and data categorization were done by the doctorate candidate and reviewed by her advisors. However, determining how publications fit them involves a lot of judgment. Thus, different results could be obtained by other researchers.

3.6 Final Considerations of the Chapter

In this chapter, we presented a secondary study that investigated ontologies in the HCI domain. The results of this study provide a panorama of research related to the topic and go deeper in some aspects of the use of ontologies to solve HCI-related problems. A total of 899 publications were considered and 35 ontologies addressing different aspects of the HCI domain were identified in 38 selected publications.

Six research questions were defined to investigate the following facets: (i) the distribution of the selected publications over the years and the type of vehicle; (ii) the HCI domain aspects addressed by the HCI ontologies; (iii) how ontologies have been used in the HCI domain; (iv) how HCI ontologies have been developed; (v) how HCI ontologies have been evaluated; and (vi) the HCI ontology quality characteristics.

According to Kitchenham and Charters (2007), the results of a secondary study help identify gaps that suggest future research. Therefore, this study contributes by providing a consolidated view of ontologies in HCI and pointing out some conclusions that can drive future research in the area. In this context, we can highlight:

(i) The adoption of ontologies in the HCI domain is a research topic that still needs to be matured. Although the first found publication is from 1994, only in the last decade ontology use became more regular in the HCI domain.

(ii) Ontologies have focused on core HCI aspects, mainly on the triad user, interface and interaction. This suggests that several HCI sub-domains have not experienced the benefits of ontology-based solutions. For instance, we did not find works using ontologies to address important HCI aspects such as prototype and user experience. Moreover, although some works deal with HCI/UI design and evaluation, there is still much to be explored. For example, we did not find any work addressing design or evaluation processes, documentation and evaluation methods. We also did not find any publication dealing with physiological computing, multiple users or smart home.

(iii) Ontologies have been used mainly in knowledge management and reasoning solutions. Considering the diversity of knowledge involved in the HCI domain and the multiple artifacts used in this context, other important solutions, such as the ones related to semantic interoperability, can be further explored.

(iv) The most sophisticated ontology-based solutions we found focus only on interface adaptation. There are many other HCI-related problems that can be addressed by using ontologies.

(v) There has been no concern with the quality of the developed ontologies. This suggests that HCI and ontology engineering professionals and researchers should work together to produce HCI ontologies. This can contribute to both areas: HCI can benefit from quality ontologies, while Ontology Engineer can find new opportunities and challenges to apply ontologies.

(vi) There has been a lack of reusing existing ontologies. Therefore, ontologies reuse in the HCI domain is an issue to be further explored. In summary, the study results show that although ontologies have been used in the HCI domain, the use has focused on only two types of ontology-based solutions and three core aspects of HCI. Moreover, HCI ontologies in general have not been developed by following ontology engineering good practices.

> Considering the advantages of using ontologies, we believe that it is worth investing efforts to improve HCI ontologies and ontology-based solutions to address problems in the HCI domain.

The found ontologies cover different (and sometimes overlapping) HCI aspects related to: user interface (UI), HCI phenomenon, user characterization, HCI/UI design, HCI/UI evaluation, context of use, accessibility, multimodality, mobile applications, pervasive environments, adaptive interactive systems, ergonomics, usability and user feedback. The ontologies have been adopted as conceptual models as well as computational artifacts. They have been used mainly to structure and infer knowledge, support reasoning, serve as a basis for developing systems (e.g., recommender systems) or other approaches (e.g., frameworks) and describe portions of the HCI domain. By analyzing the ontologies, we noticed that many of them do not exhibit important quality characteristics, which can hamper the quality of the produced ontology-based solutions. Furthermore, the HCI area has benefited from the use of ontologies in many contexts, such as knowledge management in user interface design (Suàrez; Jùnior; Barros, 2004), accessible web applications engineering (Martín et al., 2010), elicitation of user interface design recommendations (Tourwé et al., 2011), reasoning about the usability of interactive medical devices (Bowen; Hinze, 2012), modeling of user interactions in virtual reality environments (Sokolowski; Walczak, 2018), consistency between user requirements and GUI (graphical user interface) (Silva; Winckler; Trætteberg, 2019), among others.

Concerning works similar to our study, we did not find any study investigating the use of ontologies in the HCI domain. Before performing our study, we searched for secondary studies investigating ontologies in the HCI domain. Since we did not find any, we decided to carry out our study.

The study results contribute to researchers and practitioners interested in HCI and ontologies by providing information about ontologies that have been used in HCI context and how they have been used. This information allows one to reuse existing ontologies or be inspired by the ontology-based solutions to propose new ones. Furthermore, the results reveal how the ontologies have been developed, pointing out ontology engineering practices that should be considered when developing HCI ontologies. Aiming to provide advances in the state of the art, the study results reveal gaps that can be addressed in future research.

Considering the study results, we noticed that HCI ontologies could be organized in an ontology network in order to favor reuse in ontology development and increase the comprehensiveness of ontology-based solutions in the HCI domain. Therefore, we propose the Human-Computer Interaction Ontology HCI-ON, which is presented in the next chapter.

4 Human-Computer Interaction Ontology Network

This chapter presents the Human-Computer Interaction Ontology Network (HCI-ON), the knowledge framework proposed in this work to support addressing knowledge-related and semantic interoperability solutions. It refers to the artifact proposed in this work and, thus, is related to the Design cycle (Figure 1.1). It is also related to the the Rigor cycle because HCI-ON is a contribution to the knowledge base. HCI-ON was introduced in (Costa et al., 2020). Section 4.1 introduces the chapter. The HCI-ON architecture and how it was designed for assuring the necessary grounding for the networked ontologies is presented in Section 4.2. HCI-ON current version is presented in Section 4.3. HCI-ON integration with SEON is presented in Section 4.4. Section 4.5 presents mechanisms to evolve the network. Additionally, Section 4.6 discusses how HCI-ON networked ontologies have been built and integrated to the network. Section 4.7, shows how HCI-ON has been made available in a website. Last, in Section 4.8, we present some final considerations of the chapter.

4.1 Introduction

As we presented in Chapter 3, several HCI ontologies are found in the literature, representing a variety of HCI sub-domains, such as User Interface, HCI Phenomenon, User Characterization, HCI and User Interface Design, Context of Use, among others. Their use has enabled technology in a variety of initiatives such as knowledge management, reasoning, basis of a system, conceptual description, communication, semantic mapping and annotation.

The secondary study we performed (see Chapter 3) showed us that many HCI ontologies have been created and used in isolation, even when they share concepts with other existing ontologies. By reusing and integrating several ontologies, it is possible to obtain a framework that can be explored to potentialize and increase the set of solutions in the universe of discourse. Such framework would provide a more comprehensive view of the domain, enabling more robust and comprehensive solutions. Moreover, reusing existing ontologies contributes to improve the quality of the resulting ontology and increase the productivity of the ontology development process (Poveda-Villalón; Suárez-Figueroa; Gómez-Pérez, 2010). However, if there are many available ontologies, it may be hard to identify which ones are needed to solve a particular problem and integrate them to produce the necessary ontology. As a result, many times, one chooses to develop a new ontology from scratch, creating yet another ontology, even when there are others able to solve the problem of interest.

As HCI is a complex domain, we argue that in order to provide a comprehensive and consistent conceptualization for representing HCI body of knowledge, ontologies covering different HCI aspects should be integrated. However, produce a large monolithic HCI ontology is unfeasible. Suppose that there are HCI ontologies covering all HCI sub-domains. The combination of all of them would result in an ontology of the complete HCI domain. Unfortunately, building such ontology would be extremely laborious, not only due to its size, but also due to the numerous problems related to ontology integration and merging, such as overlapping concepts, diverse foundational theories and different representation and description levels, among others. Moreover, HCI sub-domains share concepts, ranging from general (e.g., Interactive Computer System, User) to more specific concepts (e.g., Usability, Quality Characteristic, and User Experience). This striking feature of the HCI domain must be considered while representing it. For achieving consistent HCI ontologies, concepts and relations should keep the same meaning in any related ontology.

Ideally, in wide domains like HCI, ontologies should not be stand-alone artifacts. Thus, we advocate that ontologies should be organized in an ontology network (ON), where ontologies are modular and related together through a variety of relationships, forming a network of interlinked semantic resources (Suárez-Figueroa et al., 2012). This way, knowledge is better structured and ontologies reuse concepts one from another, keeping consistency in shared concepts and decreasing overlap problems. By organizing ontologies in an ON, when ontologies are needed in scenarios spanning different HCI sub-domains, instead of spending effort to integrate several ontologies, one can just extract the ON portion to be used. Taking these considerations in mind, we propose the Human-Computer Interaction Ontology Network (HCI-ON) as a knowledge framework of the HCI domain.

HCI-ON was strongly inspired by SEON (Ruy et al., 2016). In this sense, we reused some elements of the SEON framework, (such as its architecture, integration mechanisms, and automatic solution to generate the ontology network specification) and we made some changes and improvements.

4.2 HCI-ON Architecture

To truly enjoy the benefits of keeping the ontologies in a network, we need to take advantage of the existing resources available in the ON for gradually improving and extending it. It is crucial to establish a sustainable architecture that supports growing the ON by adding new ontologies to it or integrating existing ontologies into it.

Ontology development and integration are not simple tasks. In addition, for building an integrated framework, it is not just putting the pieces together. Contrariwise, it is an incremental and long-term work. Therefore, the ON shall provide the means for facilitating its growth, i.e., mechanisms that deliver the high-level structures and methods for easily accommodating the new, lower-level, pieces in such a way that preserves the network properties and does not conflict with any existing part (Ruy, 2017). Finally, by organizing HCI ontologies in a network, it is not our ambition to establish a "complete HCI ontology", but providing a starting point, and a proper structure for easing the addition of new ontologies and evolution of the existing ones. Hence, HCI-ON rises with three main premises (Ruy, 2017): grounding, being based on a well-founded grounding for supporting ontology development; growth, offering mechanisms to easily building and incorporating new HCI sub-domain ontologies to the network; and consistency, promoting integration by keeping a consistent semantics for concepts and relations along the whole network.

HCI-ON adopts a three-layered architecture: in the background, we have a foundational ontology (the Unified Foundational Ontology – UFO (Guizzardi, 2005; Guizzardi; Falbo; Guizzardi, 2008; Guizzardi et al., 2013)) to provide the general ground knowledge for classifying concepts and relations in the ON; in the center, core ontologies are used to represent the general domain knowledge, being the basis for the sub-domain networked ontologies; going to the borders, well-founded and aligned domain-specific ontologies appear, describing more specific knowledge. Well-founded domain ontologies are grounded in the foundational ontology and cover HCI more relevant aspects. Aligned ontologies are existing HCI ontologies that are connected to the ON as they are. Thus, the last layer of HCI-ON, which regards domain-specific ontologies, is subdivided into two: one with well-founded domain ontologies and another with aligned domain ontologies.

Therefore, the HCI-ON architecture is organized considering the stated premises and the three ontology generality levels (foundational, core and domain), as Figure 4.1 shows. HCI-ON extends SEON architecture (Ruy et al., 2016) by decomposing the domain layer into two: one devoted to well-founded ontologies and other to accommodate aligned ontologies.

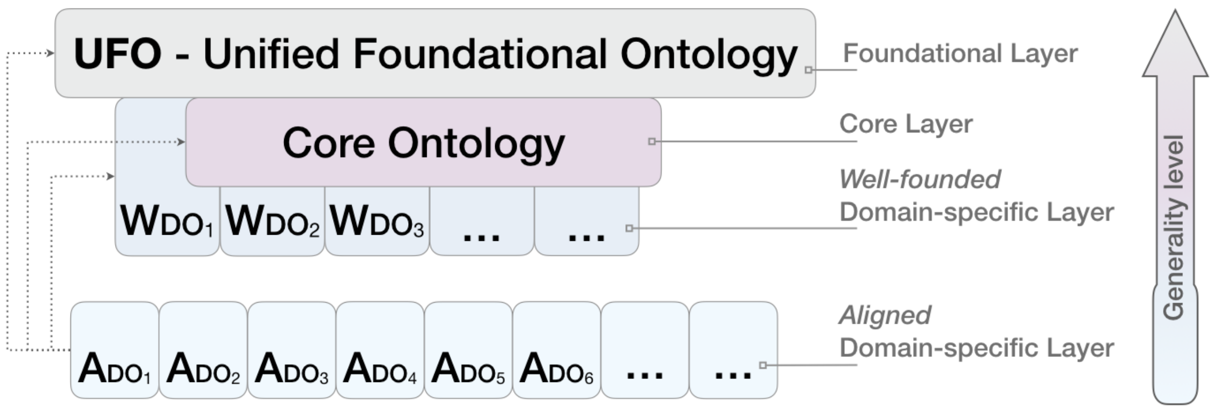


Figure 4.1 – HCI-ON Architecture (adapted from (Ruy et al., 2016)).

4.3 HCI-ON - Current Version

Figure 4.2 shows the current version of HCI-ON, containing the foundational, core and well-founded domain-specific layers. Each circle (network's node) represents an ontology. Dotted circles represent ontologies that are under development in the context of other research projects (i.e., there are concepts and relations defined for such ontologies but, at this moment, there is not a stable version of them). Arrows denote dependency relationship between networked ontologies. Dependency relationship indicates that concepts from the target ontology are reused by the source ontology.

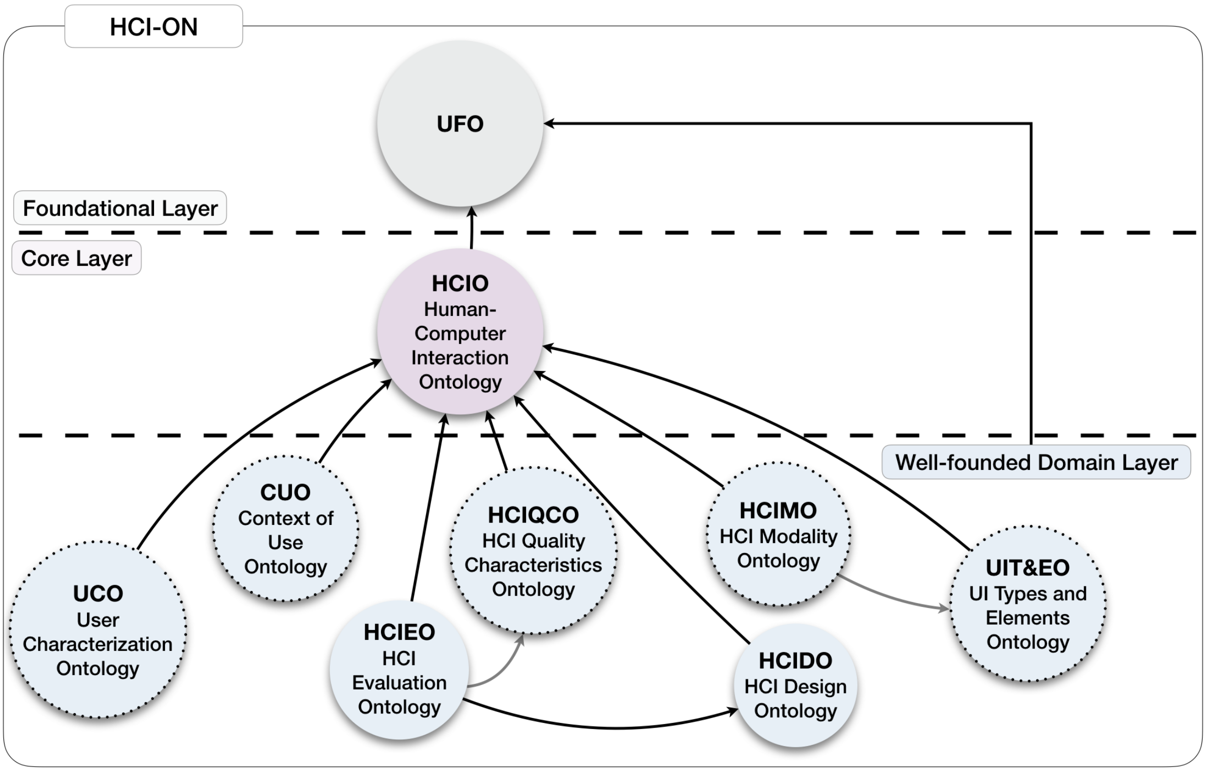


Figure 4.2 – HCI-ON current version (without the Aligned Domain Layer).

In a nutshell, the foundational layer offers the ontological distinctions for the core and domain layers, while the core layer offers the HCI core knowledge for building the domain networked ontologies. This way of grounding the ontologies in the network is helpful for engineering the networked ontologies, since it provides ontological consistency and makes a number of modeling decisions easier. It assures the grounding and consistency premises. Moreover, HCI is a very interrelated domain and, as HCI-ON increases, it has more ontologies with concepts and relations potentially reusable by the new ontologies. This reuse-based development reinforces the growth premise.

Regarding the aligned domain-specific layer, currently, HCI-ON contains the 35 HCI ontologies identified in our secondary study (see Chapter 3). These ontologies were connected to HCI-ON through alignment relationships and, thus, they are kept as they are. Therefore, they are not grounded in UFO. For better visualization, ontologies at the aligned domain-specific layer are represented in several figures. Figure 4.3, Figure 4.4, Figure 4.5 and Figure 4.6 complement Figure 4.2 and present all the networked ontologies of HCI-ON. In the figures, the id related to each ontology refer to the ones assigned to the ontologies in Chapter 3. Dotted arrows represent alignment relationships. Details on the alignment performed are presented in Section 4.6.

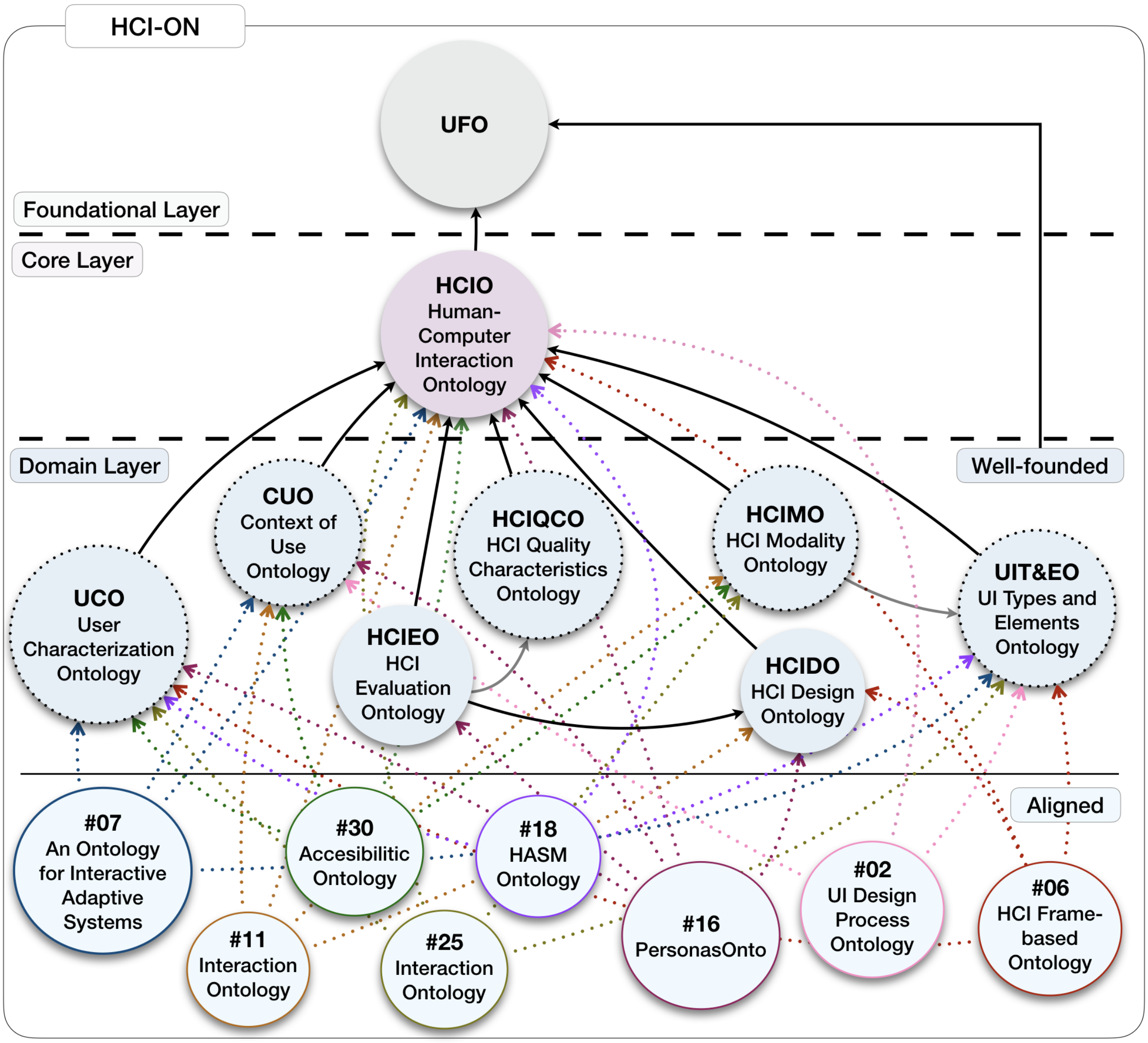


Figure 4.3 – HCI ontologies (#02, #06, #07, #11, #16, #18, #25 and #30) aligned to core and well-founded domain layers.

Next, we provide further information about each layer of HCI-ON.

4.3.1 Foundational Layer

At the top of HCI-ON, sustaining the network, is the Unified Foundational Ontology (UFO) (Guizzardi, 2005). Its ontological distinctions are used for classifying HCI-ON concepts, e.g., as objects, actions, quality, agents, roles, goals and so on. UFO provides the necessary grounding for the concepts and relations of all networked ontologies. Although UFO is incorporated as an essential part of the ON, it is not under the HCI-ON control. It was presented as a background for this research in Section 2.2.1, and is used in diverse situations henceforth.

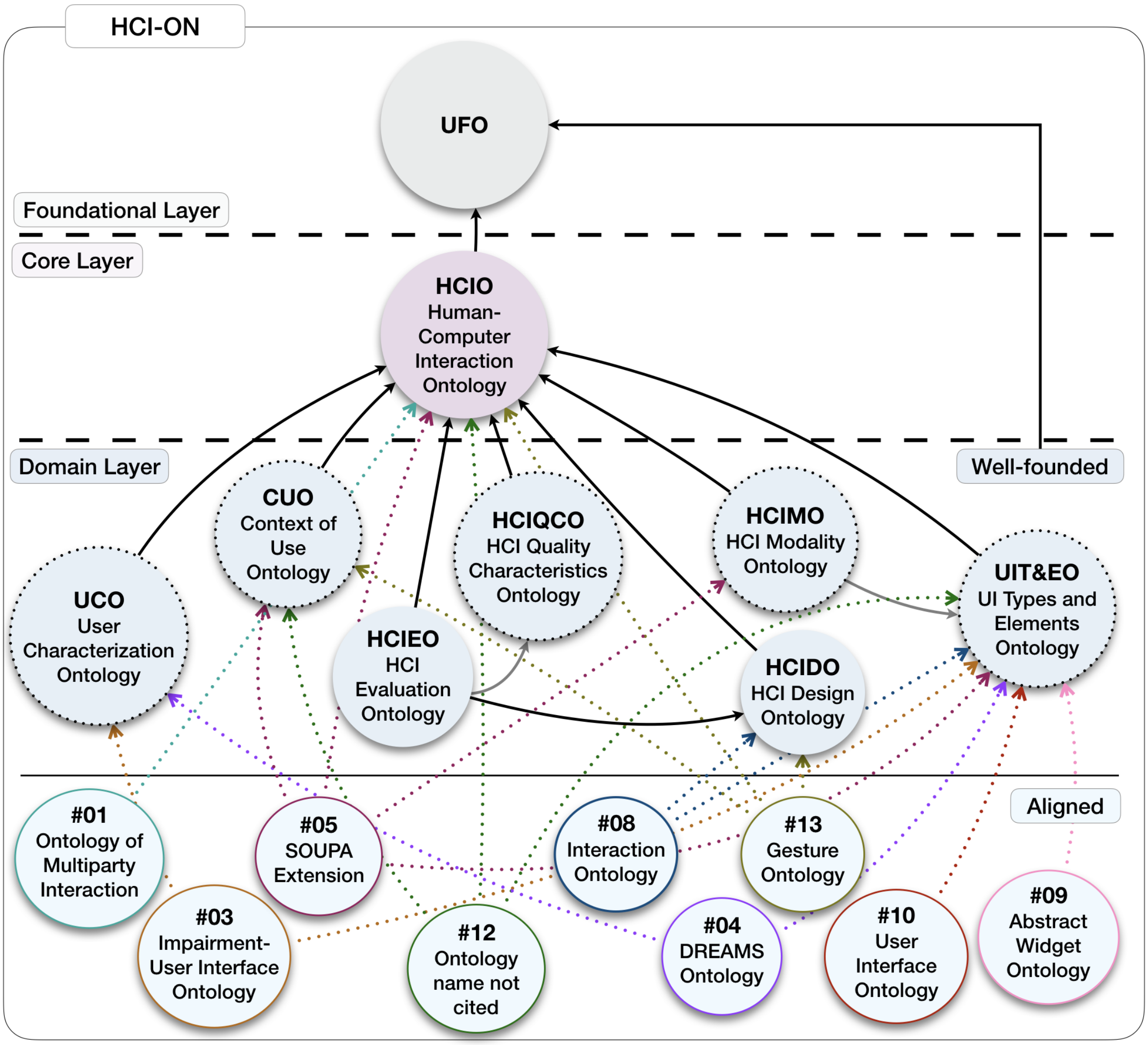


Figure 4.4 – HCI ontologies (#01, #03, #04, #05, #08, #09, #10, #12 and #13) aligned to core and well-founded domain layers.

4.3.2 Core Layer

In the center of HCI-ON, providing the HCI core knowledge for the network, there is the Human-Computer Interaction Ontology (HCIO). HCIO is a core ontology developed based on a set of HCI theories and literature. It captures that interactive computer systems have a complex artifactual nature, being constituted by software artifacts of different nature. Its purpose is to clarify the main notions and establish an explicit common and shared conceptualization about the HCI phenomenon. HCIO describes what an interactive system is, which types of actions users perform when interacting with an interactive system and, finally, what a human-computer interaction is. It is at the heart of HCI-ON, since it includes important concepts for talking about core aspects of HCI (e.g., to talk about HCI design or evaluation, it is necessary to refer to Interactive Computer System) (Costa et al., 2020). HCIO is presented in detail in Chapter 5.

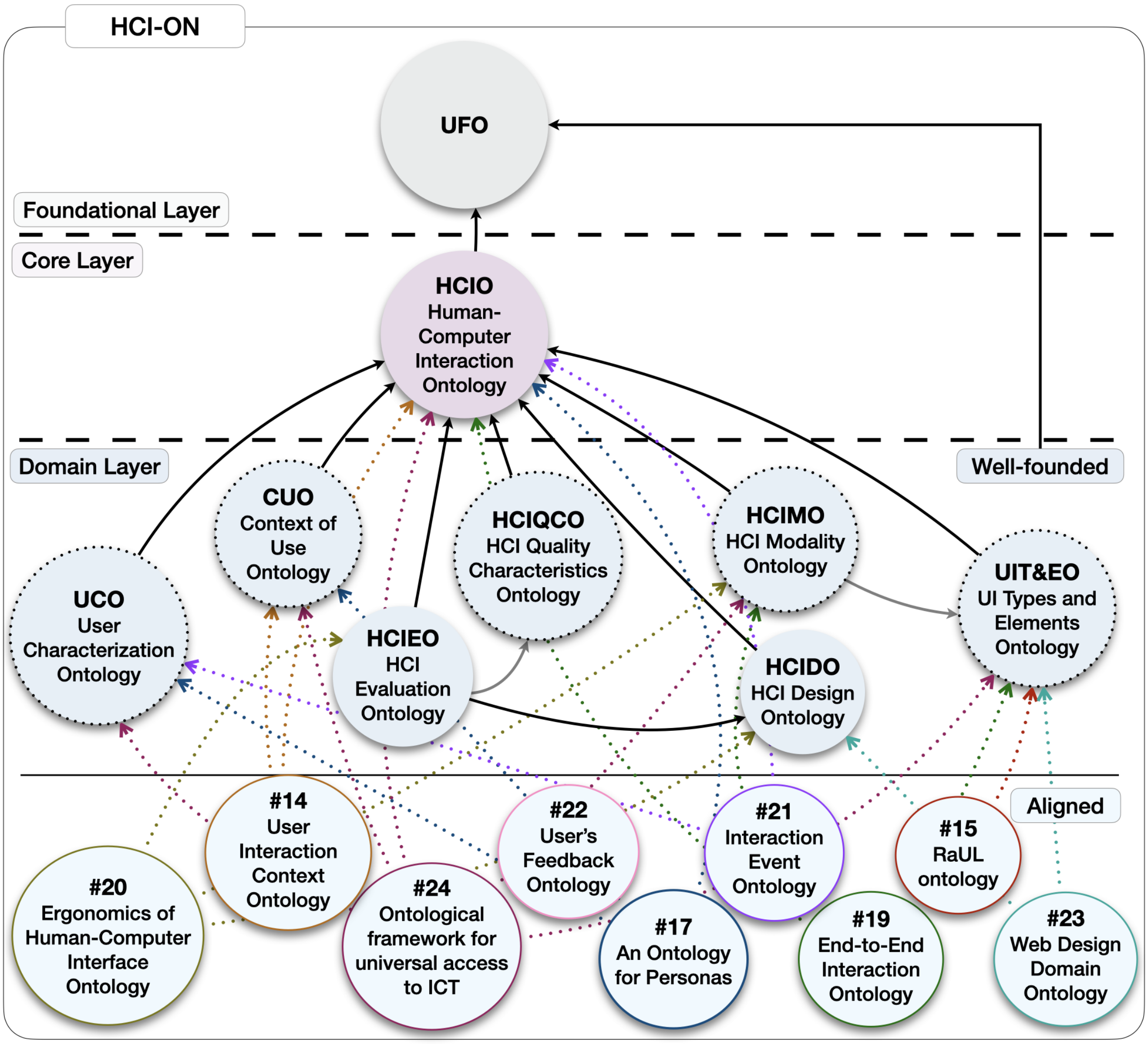


Figure 4.5 – HCI ontologies (#14, #15, #17, #19, #20, #21, #22, #23 and #24) aligned to core and well-founded domain layers.

4.3.3 Domain Layer

Under the foundational and core layers, HCI-ON places the domain ontologies. As we explained before, this layer contains *well-founded domain ontologies* and *aligned domain ontologies*, organized in two sub-layers. Well-founded domain ontologies encompass HCI sub-domains (e.g., HCI design, HCI evaluation, user interface, user characterization), are grounded in UFO and reuse concepts from HCIO. They are connected to the network through dependency relationships. Currently, there are seven well-founded domain ontologies (some of them under development): HCI Design Ontology (HCIDO) (Costa et al., 2020; Castro, 2021); HCI Evaluation Ontology (HCIEO) (presented in Chapter 6); UI Types and Elements Ontology (UIT&EO); HCI Modality Ontology (HCIMO); Context of Use Ontology (CUO); User Characterization Ontology (UCO), and HCI Quality Characteristics Ontology (HCIQCO). HCIDO and HCIEO address aspects related to, respectively, HCI design and evaluation, such as the process, produced artifacts and stakeholders, among others. HCIMO treats, in a general way, HCI styles/paradigms (modalities

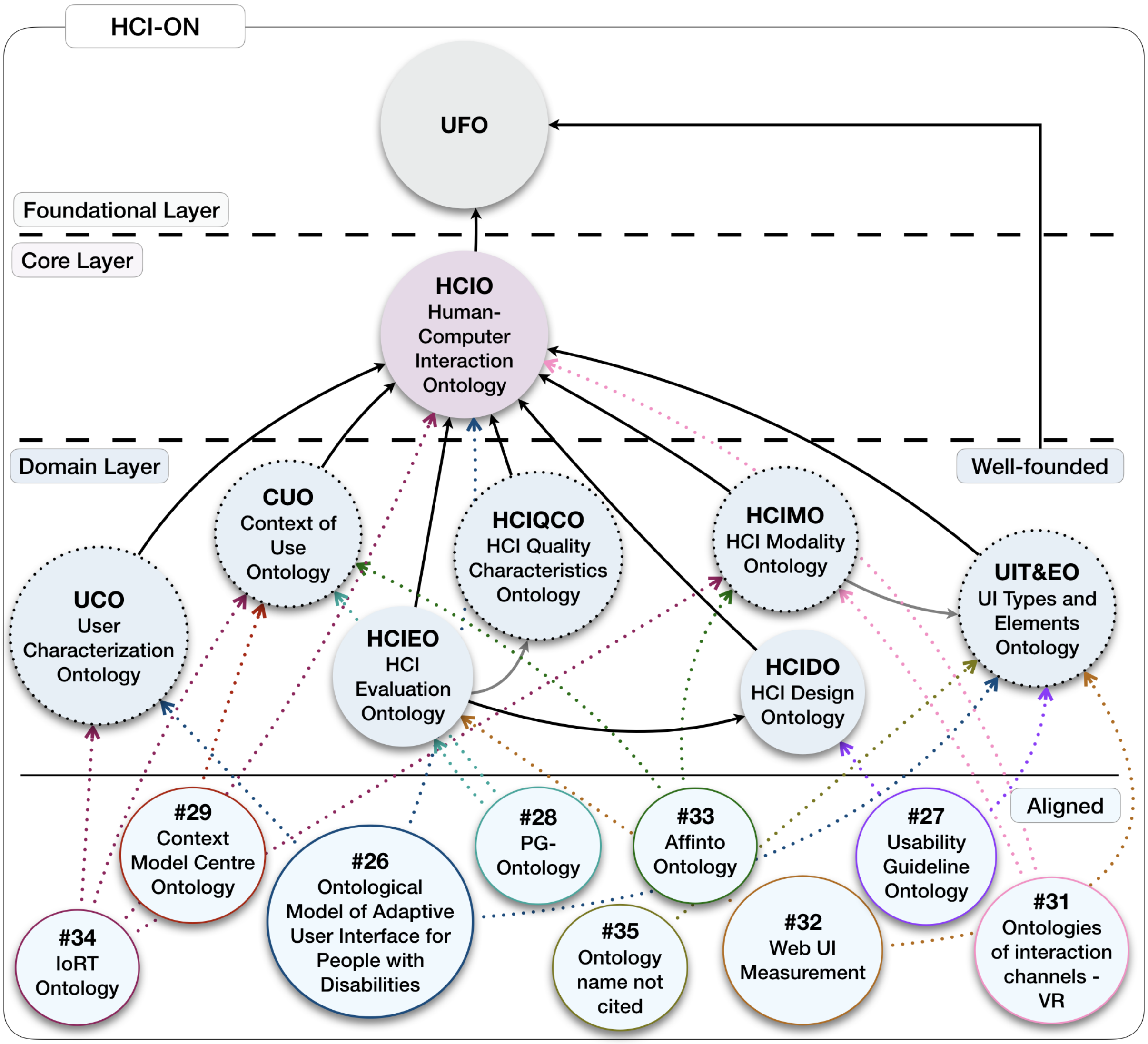


Figure 4.6 – HCI ontologies (#26, #27, #28, #29, #31, #32, #33, #34 and #35) aligned to core and well-founded domain layers.

of interaction). It connects to UIT&EO to indicate Input and Output (I/O) devices and types of interface used in these approaches. UIT&EO addresses interface types and their elements, associating them with the possible types of I/O equipment to be used in each element. CUO describes the elements that characterize a context of use, describing physical and social environments in which the interaction occurs, such as users involved in the interaction, tasks in which the interaction occurs and involved equipment (e.g., hardware, software). UCO treats, in a general way, aspects of user profile and characteristics that are important in the context of HCI domain, such as user's needs, preferences, disability, capacity, impairments. Finally, HCIQCO describes quality characteristics of user interface or interactive computer system (e.g., usability, communicability). The decision on which well-founded domain ontologies we should develop was made in order to cover relevant aspects of the HCI domain, providing knowledge to talk about the whole life cycle of an HCI project, ranging from the HCI design, user interface, modalities of interaction, HCI evaluation to context of use.

Aligned domain ontologies refer to existing HCI ontologies that are connected to HCI-ON as they are. They are not grounded in UFO and are connected to the network through alignment relationships. As we presented before, currently, there are 35 aligned domain ontologies (the ones we found in the secondary study) in HCI-ON.

4.4 Integrating HCI-ON & SEON

Since several HCI concepts are related to Software Engineering, HCI-ON is integrated to SEON (Software Engineering Ontology Network) (Ruy et al., 2016). SEON defines concepts such as software, hardware, equipment, requirements, stakeholder among others that are relevant to HCI. For example, to talk about interactive systems, user and user group and user requirements concepts addressed in SEON are necessary.

SEON (presented in Section 2.3.1) has been developed over the years and provides a well-grounded network of SE reference ontologies. It is important to highlight that as HCI-ON evolves, with new HCI networked ontologies, there is a tendency for this integration to become increasingly stronger. New HCI ontologies mean new concepts and if they are related to software engineering concepts, it is therefore of the utmost importance that SEON concepts are reused.

Both SEON and HCI-ON adopt UFO (Unified Foundational Ontology) at the foundational layer. It makes it easier to integrate them, since they keep the same conceptualization foundation and, as a consequence, contribute to consistency in the networked ontologies conceptualization.

Figure 4.7 shows the connections between HCI-ON and SEON. SEON ontologies relevant to the current version of HCI-ON are: the core ontologies System and Software Ontology (SysSwO, presented in Section 2.3.1.1), Software Process Ontology (SPO, presented in Section 2.3.1.2), the Core Ontology on Measurement (COM, presented in 2.3.1.3) (SEON core ontologies) and the Reference Software Requirements Ontology (RSRO, presented in 2.3.1.4) (SEON domain ontologies). SysSwO is reused by most HCI-ON ontologies, due to the fact that it deals with recurring and relevant aspects in HCI such as systems and software. On the other hand, SPO, RSRO and the Software Design Reference Ontology (SDRO) (Castro, 2021; Castro; Barcellos; Falbo, 2021) are reused more punctually as they support a more specific conceptualization of the HCI domain. In Figure 4.7, each circle (network's node) represents HCI-ON and SEON core or domain ontology. Dotted circles represent HCI-ON ontologies under development. Red arrows (directed arcs) represent dependency relationships from HCI-ON to SEON. HCI-ON dependency to SEON core ontologies are denoted by red solid arrows, while to SEON domain ontologies are denoted by red dotted arrows.

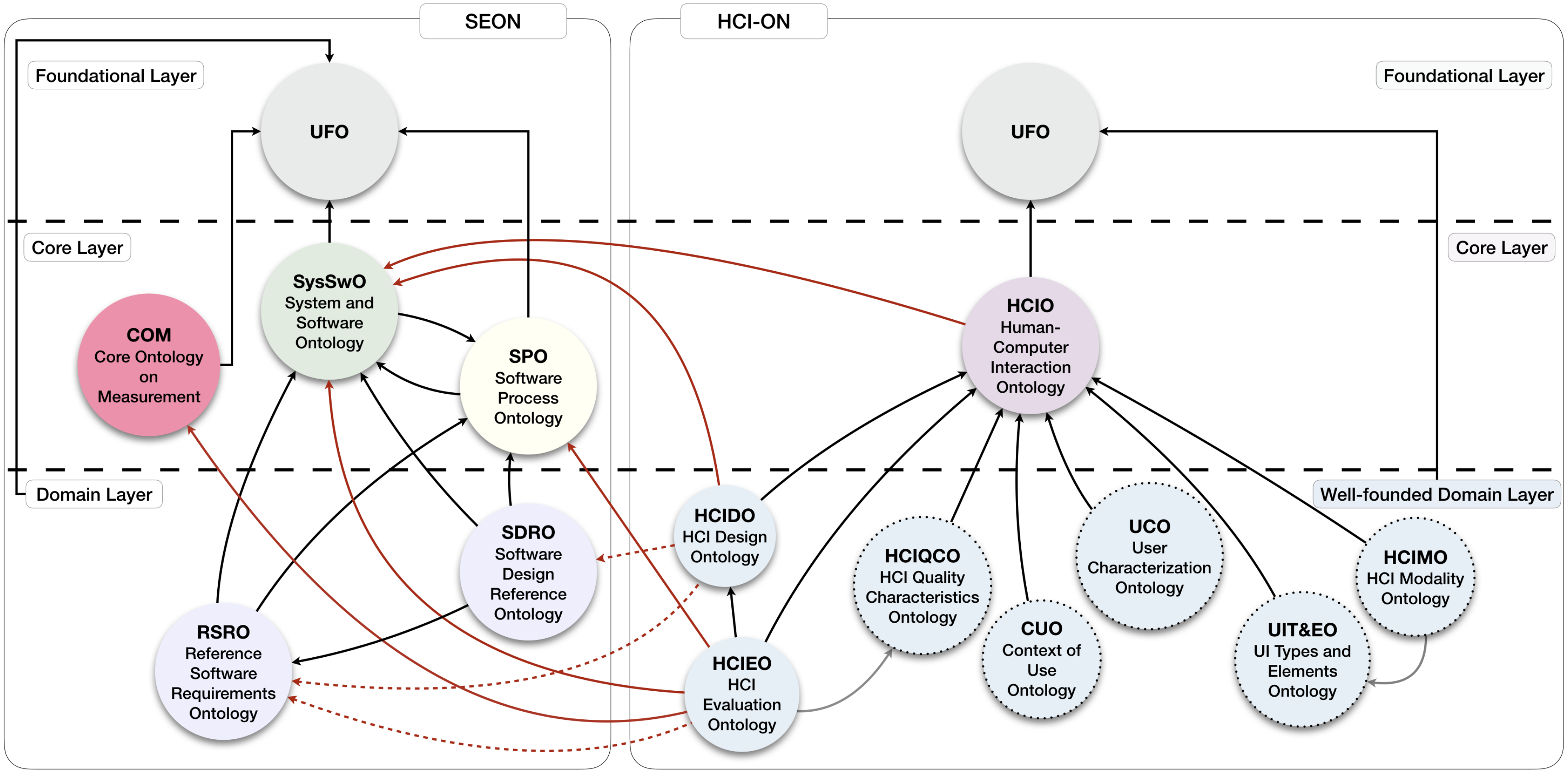


Figure 4.7 – HCI-ON relationship with SEON.

4.5 Mechanisms to support HCI-ON evolution

An ON is like a living organism, i.e., it is constantly evolving. In a network, each newly added node (i.e., ontology) contributes to the whole network. When a new ontology is added, it should reuse existing elements (from a higher or the same layer). Other ontologies, in turn, may be adapted to keep consistency, in order to share the same semantics along the whole network. Even the core ontologies can evolve to adapt or incorporate new concepts or relations discovered when domain ontologies are created or integrated. Moreover, due to the wideness and complexity of the domain addressed by the ON, it requires a continuous and long-term effort with ontologies being added and integrated incrementally (Ruy, 2017). Therefore, new ontologies can be gradually added to HCI-ON, evolving the ON and the conceptualization it provides.

Figure 4.8, illustrates the mechanisms to evolve the HCI-ON. They structure and extend the SEON's integration mechanisms (Ruy, 2017) by considering that ontologies can also be aligned or merged to the network. Before explaining the mechanisms, it is necessary to clarify the meaning of some terms used in Figure 4.8: *integrate* is the act of joining two (or more) ontologies (which deal with different subjects) using *dependency* relationships (Salamon, 2018); *merge* is the act of joining two (or more) ontologies (which deal with the same subject) also using *dependency* relationships (Suárez-Figueroa et al., 2012); *align* consists in putting two (or more) ontologies in correspondence using *alignment* relationships (Suárez-Figueroa et al., 2012); and *re-engineer* consists in changing, restructuring and improving existing ontologies (Suárez-Figueroa et al., 2012). In this work, re-engineer involves the use of UFO (i.e., the ontology is re-engineered in the light of UFO).

There are four ways (here, called mechanisms) of evolving HCI-ON: (i) develop *new* ontologies *from scratch* and *integrate* them to the network; (ii) *integrate existing* ontologies *grounded in UFO* to the network; (iii) *integrate existing* ontologies *not grounded in UFO*, to the network; and (iv) *align existing* ontologies *not grounded in UFO*, to the network. In (i), (ii) and (iii), dependency relationships are used to integrate ontologies to HCI-ON. In (iv) alignment relationships are used to put ontologies in correspondence with HCI-ON ontologies.

In (i), the ontology engineer builds a new ontology aiming at its integration to HCI-ON. The new ontology must be developed from scratch, grounded in UFO or reusing HCI-ON conceptualization (from core or well-founded domain layer, which are grounded in UFO). This ensures sharing the same foundation among all networked ontologies. The integration of the ontology to the network is done through the dependency relationships from the new ontology to networked ontologies. As an example, new HCI domains ontologies can be developed based on more general domain ontologies of HCI-ON. For instance, an ontology on Brain Computer Interface can be develop based on the User Interface Types and Elements Ontology (UIT&EO).

In (ii), the ontology engineer wants to add an existing ontology to HCI-ON and that ontology was built grounded in UFO. Thus, it is already consistent with the network foundation

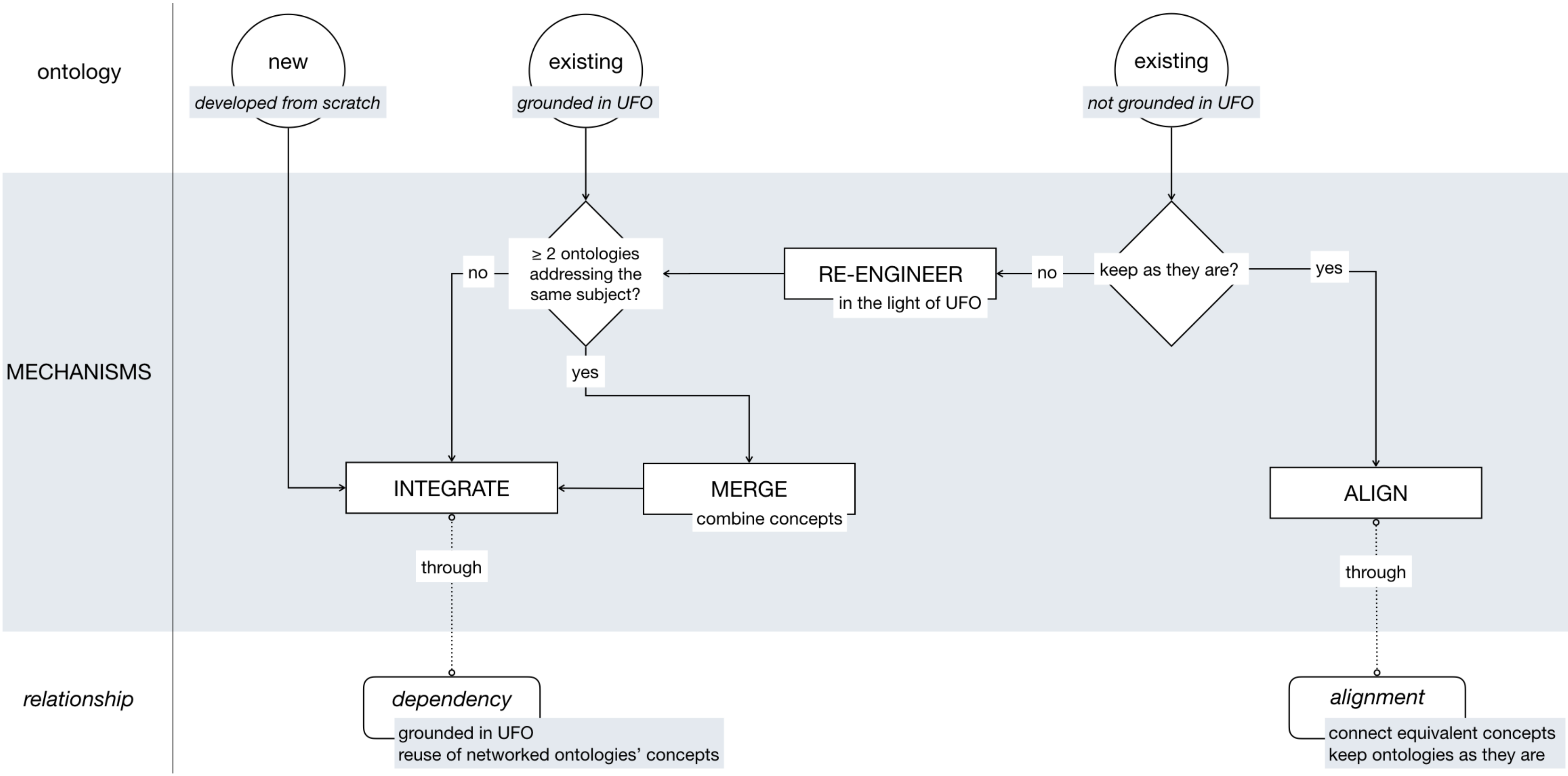


Figure 4.8 – HCI-ON evolution mechanisms.

and, thus, can be directly integrated to HCI-ON through the dependency relationships from the new ontology to networked ontologies. It may be necessary to make minor adjustments to reuse the networked ontologies conceptualization. For example, if there is an ontology about Persona developed grounded on UFO, it could be integrated to HCI-ON. When two or more existing ontologies addressing the same subject represent together the conceptualization the ontology engineer wants to add to HCI-ON (i.e., they are complementary), the ontology engineer needs to merge them before integrating them to HCI-ON.

In (iii), the ontology engineer aims to add an existing ontology to HCI-ON and that ontology was not built grounded in UFO. Thus, first, it is necessary to re-engineer the existing ontology in the light of UFO, so that the ontology will share the same basic conceptualization than the other networked ontologies and, thus, it will be possible to integrate it into the network properly. Analogous to (ii), if there are more than one ontology and they are complementary, they need to be merged before the integrations to HCI-ON. It is worth saying that the order in which re-engineer and merge are applied is at the discretion of the ontology engineer, i.e., he/she can merge the ontologies and then re-engineer them or the other way around. For example, taking the results of our SLR into account (Chapter 3), some new domain ontologies could be integrated to HCI-ON. For example, we can merge and re-engineer the ontologies #16 and #17 to produce a Persona Ontology; #07, #16, #25, #26 and #30 to produce a User Capacity and Accessibility Ontology; #06, #16, #20 and #27 to produce an HCI Procedure Ontology, and integrate them to HCI-ON.

Finally, in (iv) the ontology engineer wants to add an existing ontology to the network as it is. For that, he/she must connect the ontology with HCI-ON through alignment relationships (i.e., align the existing ontology with networked ontologies by indicating equivalences between their concepts). This mechanism relies heavily on the analysis of semantic relationships between concepts and in some cases between relationships (presented below). On one hand, aligning existing ontologies to HCI-ON makes the HCI-ON conceptualization more comprehensive. On the other hand, it allows the plugged ontologies not to be changed (not affecting applications in which they are used) and, even so, extend their conceptualizations through mappings with concepts from HCI-ON ontologies. With (iv) new existing HCI ontologies (besides the ones we found in the secondary study) can be aligned to HCI-ON.

The network evolution mechanisms should be seen as an "initial guideline" to be followed by ontology engineers to evolve HCI-ON. Regarding "guideline", we mean that the ontology engineer is provided with information that allows for a clear understanding of each mechanism and, thus, he/she can make a proper decision on each one to follow and how to follow it. As for "initial", me mean that each mechanism can (and should) be used together with other guidelines available in the literature. Moreover, the mechanisms proposed here can be further improved to incorporate other existing guidelines.

The mechanisms are not intended to restrict the use of one integration (or alignment) approach or another. Thus, the ontology engineer can choose the integration (or alignment) approach to be used (e.g., the one he/she is most familiar with). If it is not possible to adopt some existing approach, we strongly recommend that the ontology engineer performs at least mapping activities. The mapping (or correspondence) between ontologies aims to establish semantic relationships among elements (e.g., concepts, relationships) from different ontologies so that it is possible to carry out an adequate ontology integration (or alignment) (Ruy, 2017). Salamon (2018) summarizes a set of types of semantic relationships between concepts (Table 4.1) and between relationships (Table 4.2) that can help the ontology engineer in mapping activities.

Table 4.1 – Types of semantic relationships between concepts (Salamon, 2018).

Correspondence Type	**Symbol**	**Meaning**	**Example**
Equivalence	A [E] B[1]	A **is equivalent** to B. Element A represents a concept equivalent to the concept represented by Concept B.	Student [E] Undergraduate
Part of	A [P] B	A **is part of** B. Element A covers part of the concept represented by Concept B (B includes A).	Heart [P] Person
Part-Whole	A [T] B	A **is the whole of** B. Element B covers part of Concept A (B is part-whole of A).	Car [T] Chassis
Intersection	A [I] B	A **intersects with** B. Element A and Element B have properties that are common to both, but there are also properties that are either of A or of B.	Man [I] Woman
Specialization of	A [Es] B	A **is the specialization of** B. Element A represents a concept that specializes the concept represented by Concept B.	School [Es] Educational Institution
Generalization of	A [G] B	A **is the generalization of** B. Element A represents a concept that is a generalization of the concept represented by Concept B.	Organization [G] Non Profit Organization
Acts as	A [A] B	A **acts as** B. Element A represents a concept that can act as the role played by Concept B.	System analyst [A] Requirements Reviewer *(a systems analyst can play the role of Requirements Reviewer)*
Performed by	A [Dp] B	A **is performed by** B. Element A represents the concept of a role that Concept B can play.	Proof of address [Dp] Electricity bill
No existing relationship	A [-]	A **has no relationship**. Element A represents a concept that has no relation to any concept B.	Nurse [-] *(the concept is present in one model and has no relation to any other concept in another model)*

Table 4.2 – Types of semantic relationships between relationships (Salamon, 2018).

Correspondence Type	Symbol	Meaning	Example
Equivalence	R1 [Eq] R2	R1 **is equivalent to** R2. The relation R1 has the same meaning as the relation R2. Furthermore, the concepts involved in R1 are equivalent to the concepts involved in R2.	Student **fulfills** Discipline. Student **studying** Discipline.
Specialization of	R2 [Esp] R1	R2 **is the specialization of** R1. The relationship R2 represents a relationship between more specific concepts, while R1 is a relationship between more generic concepts. Thus, the concepts involved in R2 are specializations of the concepts involved in R1.	Education Professional **works in** an Educational Institution. University Professor **works at** University.
Generalization of	R1 [Gen] R2	R1 **is the generalization of** R2. The relationship R1 is a relationship between more generic concepts, while R2 represents a relationship between more specific concepts. Thus, the concepts involved in R2 are specializations of the concepts involved in R1.	Student **studies at** an Educational Institution. University **studies at** the University.
Inverse of	R1 [Inv] R2	R1 **is the inverse of** R2. The relation R1 has the inverse meaning of the relation R2. The source and target concepts of relationship R1 must be equivalent to the concepts of target and source of relationship R2.	Professor **teaches** Discipline. The course **is taught by** Professor.
Derivation	R [De] R1...Rn	R **is the derivation of** R1,... Rn. R represents a derivation of other relations. The origin concept of R must be equivalent to the origin concept of R1 and the destination concept of R must be equivalent to the destination concept of Rn.	R1: Analysis **evaluates** Data. A2: Data **is collected in** the Sample. R3: Analysis **characterizes** the Sample. R3 is a derivation of R1 and R2.

4.6 Applying the mechanisms of evolution to build the HCI-ON

We have developed the HCI-ON ontologies shown in Figure 4.2 by following the mechanisms presented in the previous section. Hence, it was possible to validate and refine them.

The first ontology we built was HCIO, at the core layer. We developed HCIO and integrated it to HCI-ON by following the first mechanism (i.e., develop new ontologies from scratch and integrate them into the network) and reusing concepts from SEON through dependency relationships. Then, we added the domain ontologies at the well-founded domain-specific layer, connecting them (through dependency relationships) to HCIO, to each other and to SEON ontologies, according to their scope. HCIO and the well-founded domain ontologies were developed from scratch, based on knowledge extracted from the literature and International Standards, and grounded in UFO. HCIO conceptualization has been important to develop the well-founded domain ontologies, since its concepts have been reused to build them. For example, the *User*, *Interactive Computer System* and *Human-Computer Interaction* HCIO concepts were reused by all well-founded domain ontologies (HCIO is presented in detail in the Chapter 5). Hence, by providing the HCI core concepts, HCIO is essential to the ON growth. For example, in the well-founded domain ontologies layer, by extending the HCIO's User Interface concept, we are able to address different user interface types and elements as well as some aspects of interaction modality. Thus, we evolved the network by addressing these aspects in the UI Types and Elements Ontology (UIT&EO) and HCI Modality Ontology (HCIMO), respectively. When applying the first mechanism (i.e., develop new ontologies from scratch and integrate them into the network), HCI-ON well-founded domain-specific ontologies also helped to develop other well-founded domain-specific ontologies. For example, HCIEO reuses HCIQCO concepts (HCIEO is presented in detail in Chapter 6).

We also added ontologies to HCI-ON by following the fourth mechanism (i.e., align existing ontologies not grounded in UFO, to the network) to add to HCI-ON, through alignment relationships, the ontologies we found in the literature. Concepts from existing ontologies were aligned with concepts from well-founded domain-specific ontologies of HCI-ON (as shown in figures 4.3, 4.4, 4.5 and 4.6). By doing that, HCI-ON conceptualization gets more comprehensive by including the aligned ontologies conceptualization, while the aligned ontologies can benefit from the HCI-ON conceptualization as a whole. Since the aligned ontologies were kept as they are (i.e., not re-engineered), they are not grounded in UFO.

Briefly, to perform the literature ontology alignment, we searched and selected literature ontology concepts related to the HCI domain addressed by the ontologies of the network. Then, we performed semantic mappings between the selected and ONs concepts, i.e., we checked the equivalence (Equivalence type, Table 4.1) between concepts. We also verified the conceptual or operational model (the available ones) to contextualize the meaning of the concept, aiming to reach a more accurate confirmation of the semantic mappings. When the concept's definition was made available, we examined it and its contextualization in the model. When there were only terms (concept without definition), we examined its contextualization in the model. It is important to note that this procedure has not been performed for all concepts/terms of each literature's ontology for the following reasons: there were ontologies with numerous concepts/terms; there were concepts/terms that did not refer to the HCI domain, but the application domain; and the alignment (semantic mapping) of just one concept/term to a network concept is enough to establish a connection. Due to a large number of literature ontologies concepts, and relationships, we did not perform the relationship alignment (Table 4.2). In addition, as the domain to be treated by network ontologies that are still under development has already been previously studied and analyzed, an idea of the concepts necessary for the description of the domains is already available. Therefore, the alignment performed to these ontologies was possible.

Concerning the second (i.e., integrate existing ontologies grounded in UFO to the network) and third (i.e., integrate existing ontologies not grounded in UFO, to the network) mechanisms, we have not applied them yet. However, in the context of (iii), we have already used HCIO to identify mappings between concepts of different ontologies and help us merge and re-engineer existing ontologies to integrate the resulting ones into HCI-ON ontologies. To illustrate the use of HCIO with this purpose, Table 4.3 shows semantic mappings of equivalence type between concepts from existing ontologies identified in our secondary study and HCIO.

4.7 HCI-ON Specification

Building an Ontology Network involves various aspects regarding the creation, integration and evolution of the networked ontologies. Providing an effective access to the network content is essential for its application and improvement. An ON should be presented considering different perspectives and providing useful information. When working with reference ontologies, the ontology diagrams should be accompanied with further information about their concepts, relations and other connections. Moreover, the network documentation shall be kept always accessible and updated. Documenting an ontology's specification is a laborious work which increases when considering ontologies in a network (Ruy, 2017).

Manually keeping all the ON information available in an accessible format is a complex task. Thus, inspired by SEON specification (<dev.nemo.inf.ufes.br/seon/>), we reused and adapted the transformation tool proposed by Ruy (2017), which is able to collect data from ontology models built using Astah[2] and transform it into an HTML specification.

The code also performs network consistency checks on networked ontologies' concepts (i.e., definition; ground; relationships; and ontology source) and produces a preliminary operational version of HCI-ON and SEON in OWL. The adapted code used to create HCI-ON website is available at <https://github.com/sd-costa/HCI-ON_websitecode>.

HCI-ON specification is available as a website at <dev.nemo.inf.ufes.br/hcion/>. HCI-ON website aims to provide useful information about the network (i.e., description, architecture, evolution mechanisms, among others) and its networked ontologies. Figure 4.9, shows the home page and the website's navigation bar (top of the page).

Figure 4.10 shows the page that present detailed information of the networked ontologies, such as their description, related ontologies, models/diagrams, concepts definition, detailing of concepts. Each of the networked ontologies has a page like the one shown in Figure 4.10.

There are also other features, such as: Searcher: a search engine for finding concepts by name and definition (Figure 4.11a); Graph: the network visualization as a graph; Stats: some network statistics (Figure 4.11b); Videos: some videos about HCI-ON; and Authorship Info: ontology engineers responsible for ontology development, and experts who participated in its

[2] <https://astah.net/>

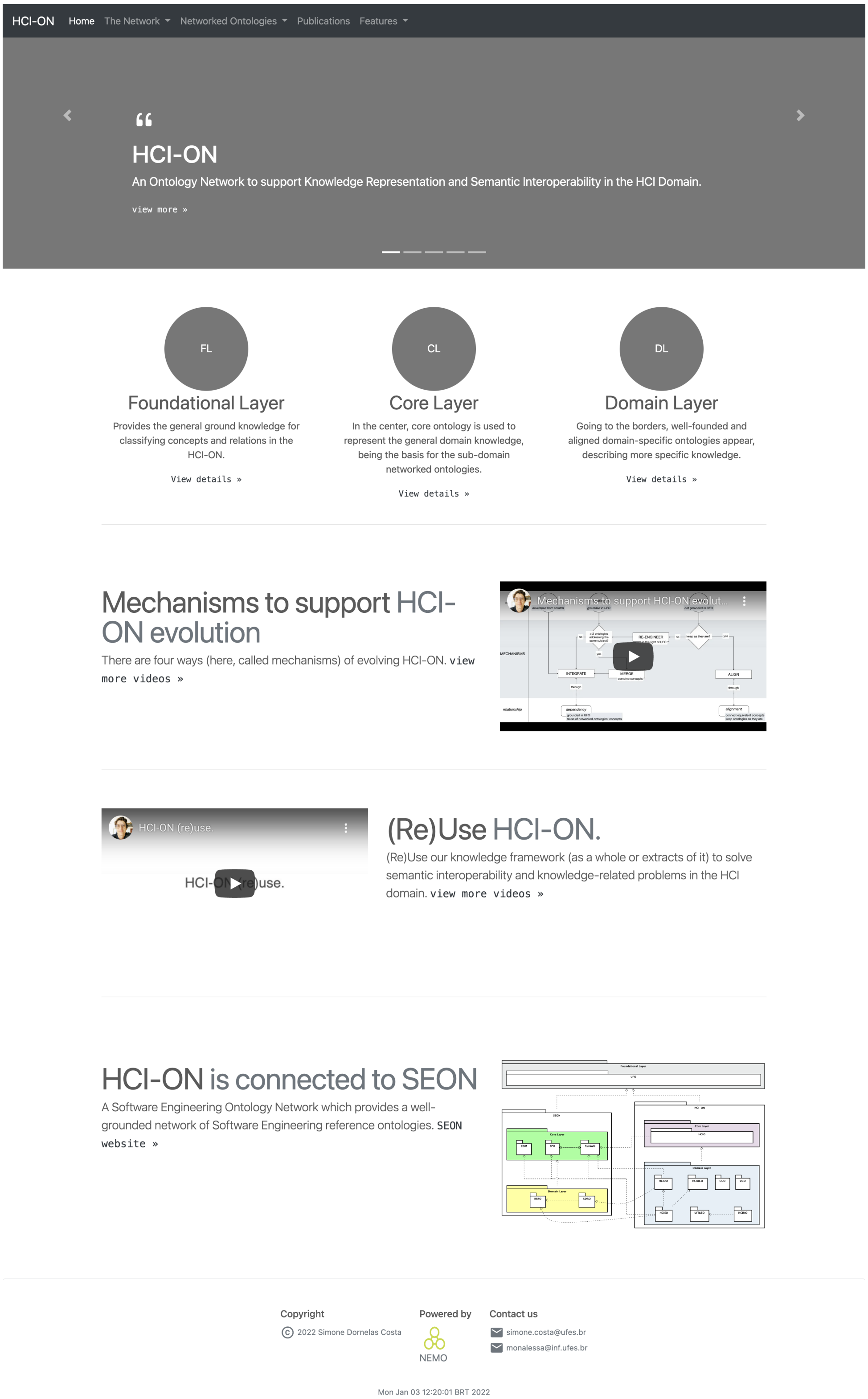


Figure 4.9 – HCI-ON's home page.

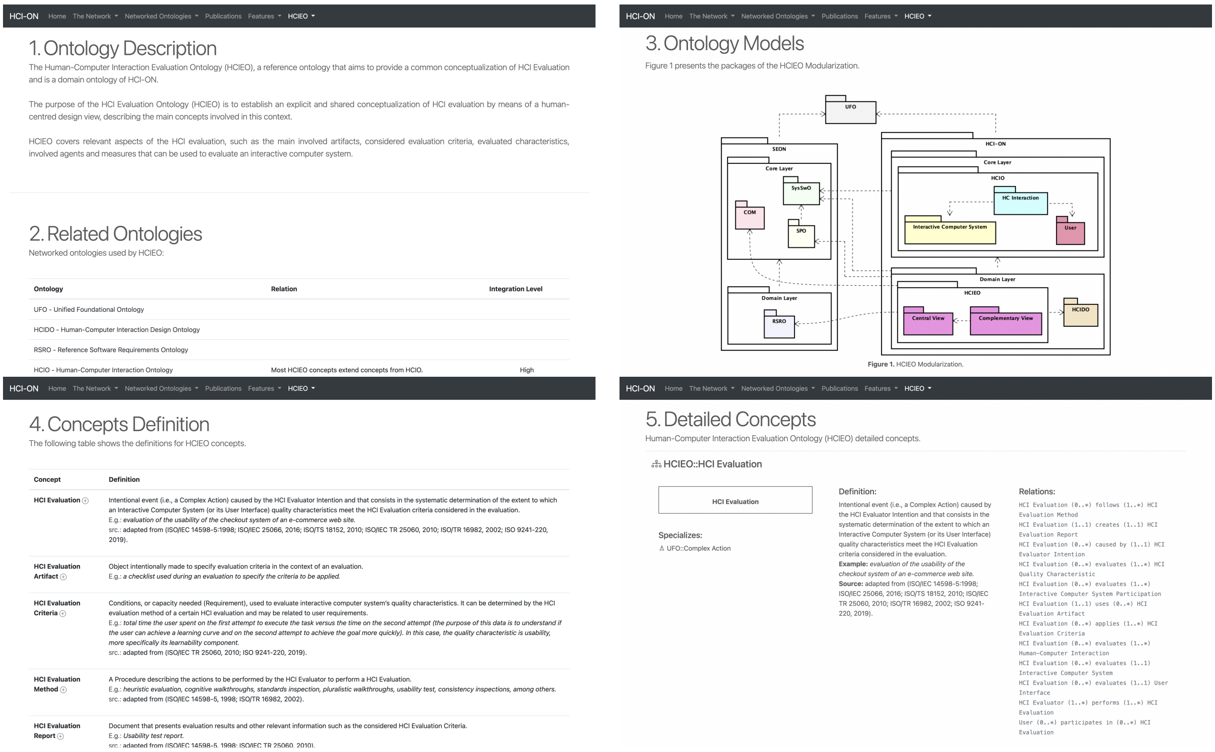
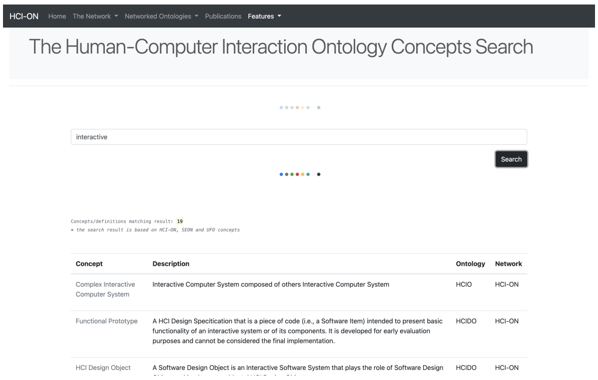


Figure 4.10 – Networked ontology's page.

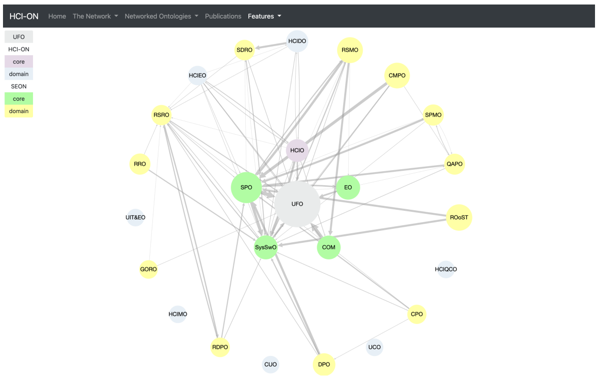
Table 4.3 – Aligning ontologies using HCIO.

Id	HCIO concepts								
	User	Human-Computer Interaction	Interactive Computer System	Interactive Software System	User Interface	Input Equipment	Output Equipment	User Goal	User Participation
#01	User	Interaction, Dialogue			Interface				
#02	Agent	Interaction, Dialogue	Interaction Object						Action
#05					I/O Device	Input Device	Output Device		
#06	Target User			Software					
#07		Interaction							
#11	User		Device	Application		Input Component	Output Component		
#12			Device						
#13	User		Device						
#14	User								Action
#16	Person							Goal	
#17	Person							Goal	
#18	Users	User-interaction-with-a-haptic-device	Haptic-Devices						
#19	User	Interaction			Interface				
#21	Person	Interaction Event							
#24	User		Device		User Interface				User Action
#25	Craftswoman					Input Medium	Output Medium		
#26	User		Device		Interface				
#29			Device						
#30	Users								Activity Participation
#31						Physical Input, Mouse, Keyboard	Physical Output		

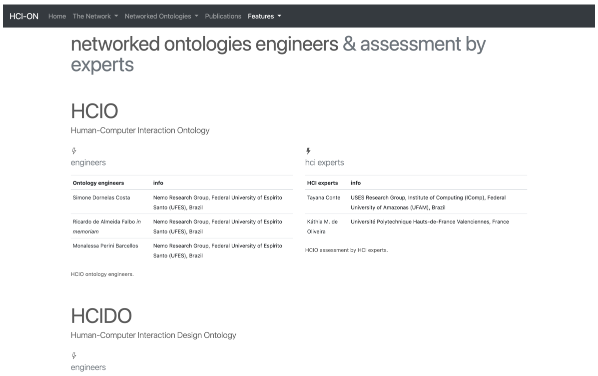
evaluation (Figure 4.11c). There is also a page that lists all the publications related to HCI-ON, as shown in Figure 4.11d.



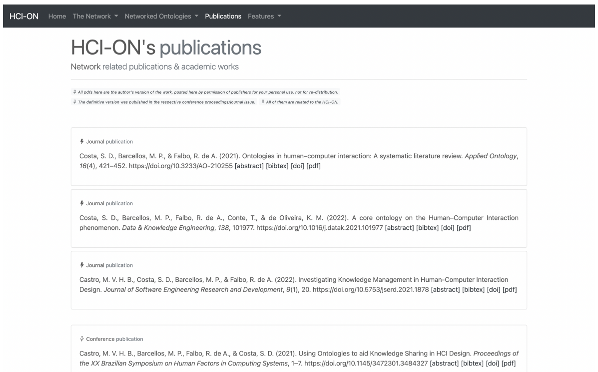
(a) Searcher page.



(b) Graph page.



(c) Authorship Info page.



(d) Publication page.

Figure 4.11 – Some Features & Publication's pages.

HCI-ON website aims to ease understanding, using, and improving HCI-ON, especially for people from other research groups and industry. It also allows a faster and reliable way to publish HCI-ON new versions. More than access, HCI-ON Specification (website) is an important instrument for the network evolution.

4.8 Final Considerations of the Chapter

This chapter presented HCI-ON, a Human-Computer Interaction Ontology Network designed aiming at: (i) establishing a layer-based structure for supporting ontologies addressing different HCI aspects; (ii) providing an effective support for integrating and aligning domain ontologies; and (iii) being applied for solving knowledge-related and semantic interoperability problems in HCI.

Although achieving all these objectives is a long-term effort, we believe that the current version of HCI-ON represents a starting point in this direction. The current results of the network (architecture, networked ontologies and evolution mechanisms) are crucial points to achieve its purpose and evolution. We realize that the growth of the network provides benefits such as: the network becomes more consistent and able to support the integration of new and improvement of current ontologies.

HCI-ON is the HCI reference knowledge framework proposed in this thesis to reduce semantic interoperability and knowledge-related problems in HCI domain. Currently, HCI-ON is being used in some applications, which will be discussed in Chapter 7. In this chapter we presented the current version of HCI-ON. We expect to evolve HCI-ON, making it more comprehensive by adding new domain ontologies, advancing with the defined mechanisms and exploring the use of HCI-ON in practical situations. Moreover, for HCI-ON to reach other interested people, it was made available at <http://dev.nemo.inf.ufes.br/hcion/>.

5 Human-Computer Interaction Ontology - HCIO

This chapter presents the Human-Computer Interaction Ontology (HCIO), the reference ontology that aims to provide a common conceptualization of the HCI phenomenon and serves as a core ontology of HCI-ON. As part of HCI-ON – the artifact proposed in this research – it is related to the Design cycle (Figure 1.1). It was developed to enable HCI-ON to satisfy R3 (address core aspects of the HCI phenomenon). As HCI-ON, it is also related to the Rigor cycle, since it contributes to grow of the existing knowledge base. HCIO was published in (Costa et al., 2022). Section 5.1 introduces the chapter and presents two real cases of HCI phenomenon that are later used to demonstrate that HCIO is able to represent real world situations. Section 5.2 presents HCIO, its architecture and modularization into sub-ontologies and describes each of them. Section 5.3 discusses how we evaluated HCIO. Section 5.4 makes some discussions about HCIO conceptualization. Section 5.5 compares HCIO and existing ontologies addressing HCI phenomenon. Last, Section 5.6 closes the chapter.

5.1 Introduction

As argued in the previous chapter, in an ontology network, core ontologies provide concepts that are common to the addressed domain and cross several sub-domains. When analyzing the 35 ontologies found in the secondary study (see Chapter 3), we noticed that, although some core concepts should be common to them, there are inconsistencies among the ontologies conceptualization, even concerning core concepts. Even concepts from ontologies covering the HCI phenomenon are not consensual. We believe that, as we discussed before, this is mainly due to the fact that most of the ontologies were developed to solve specific problems for the purpose of practical applications, in specific contexts. Moreover, most of the ontologies do not present the concepts clearly. This demands interpretation from the reader, which is susceptible to misunderstanding. Thus, we decided to develop the Human-Computer Interaction Ontology (HCIO), to properly represent the HCI phenomenon, serve as a reference to the HCI domain and provide core concepts to HCI-ON networked ontologies.

Before presenting HCIO, next, we describe two scenarios of use of human-computer interaction, which are used later to exemplify (i.e., instantiate) HCIO concepts. The cases were performed in the real world and here they are presented by means of storyboards using fictious names. In the first case, a person (John) interacts with his desktop computer to quote flights prices on the Internet. In the second case, a person (Rino) interacts with his smart watch to monitor his performance in a run. Figure 5.1 illustrates the case where John quotes flights costs on the Internet. John is a New Yorker student who intends to attend a conference in Rio de Janeiro, Brazil.

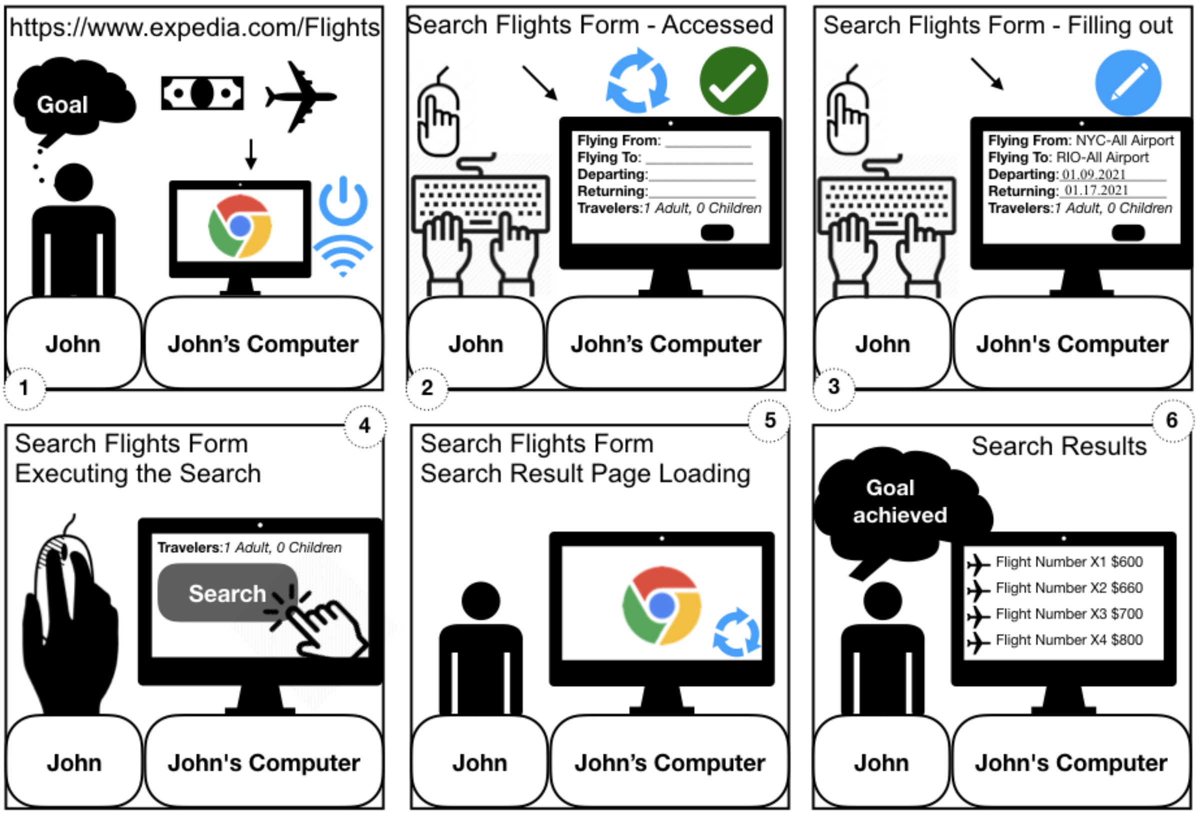


Figure 5.1 – Storyboard: John searching the Internet to quote flights prices.

John's desktop computer is ready to start the interaction (Figure 5.1, 1st picture), i.e., the computer is properly connected to the Internet and with Chrome running. John accesses Expedia's site (<https://www.expedia.com/Flights>) and faces the form "Search Flights" (2nd picture). He notices that some fields of the form appear filled, while others are blank. John fills out the required fields (3rd picture). After that, John executes the search by clicking with the mouse the "Search" button, activating it (4th picture). John notices that Chrome is loading (5th picture) and, seconds after, he gets a new page with the list of flights and corresponding prices and he identifies the flight with lowest price, achieving his goal (6th picture).

Figure 5.2 illustrates the case where Rino interacts with his smart watch to monitor performance in a run.

Every day, Rino runs for 30 minutes. Aiming to monitor his heart rate and performance, he interacts with his Apple watch (Figure 5.2, 1st picture). Before starting to run, Rino presses the watch-crown that triggers Siri[1] (2nd picture), which asks "What can I help you with?" (3rd picture). Rino says "Start outdoor run" (4th picture) and the watch opens the Activities app. To set the running time, Rino touches the widget [...] (5th picture) and then the time widget (6th picture). Rino touches the widget [+] until it reaches 30 minutes (7th picture) and then touches

[1] <https://www.apple.com/siri/>

Figure 5.2 – Storyboard: Rino monitoring his performance in a run.

the start widget (8th picture). The watch shows a countdown (9th picture) and Rino starts running. After a few minutes he turns his wrist to check his heart rate and performance (10th picture). He sees the following data: 10:22, 18 of running, heart rate at 169bpm, distance 0.91mi and pace 4'66"/mi. Rino continues running. After 15 minutes, Rino feels the watch vibrate, turns his wrist and sees a message informing that he has completed half of his way (11th picture). He continues running until he feels the watch vibrate again. He looks at the watch and sees a message informing that he has ran 30 minutes and thus his goal has been achieved (12th picture).

5.2 The Human-Computer Interaction Ontology (HCIO)

The purpose of the Human-Computer Interaction Ontology (HCIO) is to establish an explicit and shared conceptualization of the HCI phenomenon, describing the main concepts involved in this phenomenon. The knowledge sources used to build HCIO include standards, such as (ISO, 2019a), theories (Norman, 1986; Souza, 2005), models (Abowd; Beale, 1991; Hewett et al., 1992) and relevant literature in the HCI area, such as (Carroll, 2014), (Sutcliffe, 2014), (Preece; Sharp; Rogers, 2015), (Dix et al., 2004), (Benyon, 2010), (Norman, 2013), (Krol et al., 2016), (Fairclough, 2009), (Norman, 2009), (Fairclough; Gilleade, 2014), (Zander; Krol; Gramann, 2018), (Belkhiria; Peysakhovich, 2020), (Zander et al., 2010), (Clites et al., 2018), (Zander; Kothe, 2011), (Nielsen, 1993), (Oliveira et al., 2017), (Rogers; Sharp; Preece, 2011), (Saffer, 2010), (Callan et al., 2016), among others. HCIO also includes knowledge obtained from the study of the SLR ontologies (we did not reuse the ontologies themselves because, as we explained in Section 3, they have several limitations). It is worth pointing out that understanding the different types of interaction (see Section 2.1.1) was very important to ensure that HCIO is capable of representing all of them. When we built the first version of HCIO, we did not consider different types of interaction. As a result, that version was not able to represent several interaction scenarios that occur in the real world. Domain experts evaluated the ontology and pointed out these problems. To solve them, we incorporated to HCIO knowledge of different types of interaction.

As we said in the previous chapter, HCIO is grounded in UFO (Guizzardi, 2005) and reuses concepts from the core System and Software Ontology (SysSwO) (Bringuente; Falbo; Guizzardi, 2011; Duarte et al., 2018). Given that a human-computer interaction involves communication between user and interactive computer system, HCIO is composed of sub-ontologies to deal with the interaction participants and with the interaction itself. The *Interactive Computer System* sub-ontology addresses what an interactive computer system is and its elements, including the user interface. The *User* sub-ontology focuses on the user and its possible actions when interacting with an interactive computer system. Finally, the *HC Interaction* sub-ontology links concepts from the other two sub-ontologies to define what a human-computer interaction is. Figure 5.3 shows HCIO sub-ontologies and its relations with SysSwO and UFO. In the figure, dependency relations mean that the source ontology reuses concepts from the target ontology.

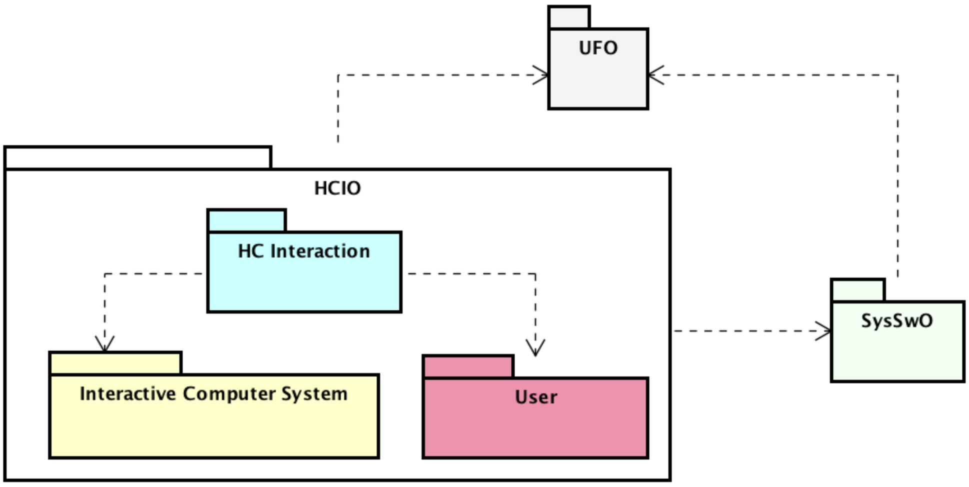


Figure 5.3 – HCIO architecture.

HCIO was developed by following SABiO (Falbo, 2014), which prescribes that the ontology scope must be defined by means of competency questions, i.e., questions the ontology must be able to answer and are used as a basis to develop the ontology conceptual model. To that end, we considered the basis of HCI previously presented (see Section 2.1.1) saying that **human-computer interaction** is the communication process that occurs during the use of an **interactive computer system** and that involves **user** actions. As consequence, to cover the HCI phenomenon we should focus on concepts related to interactive computer system, user and HCI itself. Table 5.1 shows HCIO competency questions (CQ). CQ01 to QC04 help understand what an interactive computer system is and its software and hardware elements. QC05 to QC09 are to understand user, the actions he/she performs when interacting with interactive computer systems and what causes user to interact with the system. CQ10 to Q14 refer to the human-computer interaction itself, addressing user inputs, system outputs, and actions and interpretations involved in the interaction. QC15 is about goal achievement, which is relevant when the user interacts with the system aiming to achieve a certain goal.

Next, we present HCIO following its modularization into sub-ontologies. In the conceptual models, we kept the colors used in Figure 5.3 to identify the concept source. In the models description we refer to the scenarios of use presented in Section 5.1 to exemplify HCIO concepts.

Table 5.1 – HCIO Competency Questions.

Focus	**Id**	**Description**
Interactive Computer System	CQ01	What is an interactive computer system?
	CQ02	What is an interactive software system?
	CQ03	What is a complex interactive computer system?
	CQ04	What does make up the user interface of an interactive computer system?
User	CQ05	What is a User?
	CQ06	How can a user interact in human-computer interactions?
	CQ07	Considering intentionality, how can a user interact with an interactive computer system?
	CQ08	Why does a user intentionally interact with an interactive computer system?
	CQ09	What does make up a complex user participation?
HCI	CQ10	What is a human-computer interaction?
	CQ11	Considering the human-computer interaction, how can a user participation cause another user participation?
	CQ12	How is a user input processed by an interactive computer system?
	CQ13	How does a user receive an output from an interactive computer system?
	CQ14	How is a user input processed by an interactive computer system and how is the corresponding output presented to him/her?
	CQ15	How does a user evaluate if his/her goal was achieved in a human-computer interaction?

5.2.1 Interactive Computer System Sub-Ontology

This sub-ontology aims to answer CQ01 to CQ04 and, thus, focuses on defining interactive computer system. It is mainly an extension of SysSwO (see Section 2.3.1.1). Figure 5.4 shows the conceptual model of the Interactive Computer System sub-ontology. In the figure, we used doted lines to separate the ontologies in layers. At the top, we have UFO, providing the general foundation. At the center, there is SysSwO, containing core concepts related to computer systems. At the bottom, there is HCIO concepts, grounded in UFO or specialized from SysSwO (which is also grounded in UFO). Relations that ground SysSwO or HCIO concepts in UFO (i.e., specializations) are shown in blue. In the text, we use *italics* to refer to UFO concepts, <u>*underline italics*</u> to SysSwO concepts and ***bold italics*** to HCIO concepts. In the models description, we present some of the axioms defined to address constraints not captured in the models.

Interactive Computer System is a subtype of <u>*Computer System*</u>, and like the latter, it combines hardware and software. Concerning hardware, an ***Interactive Computer System*** is (the

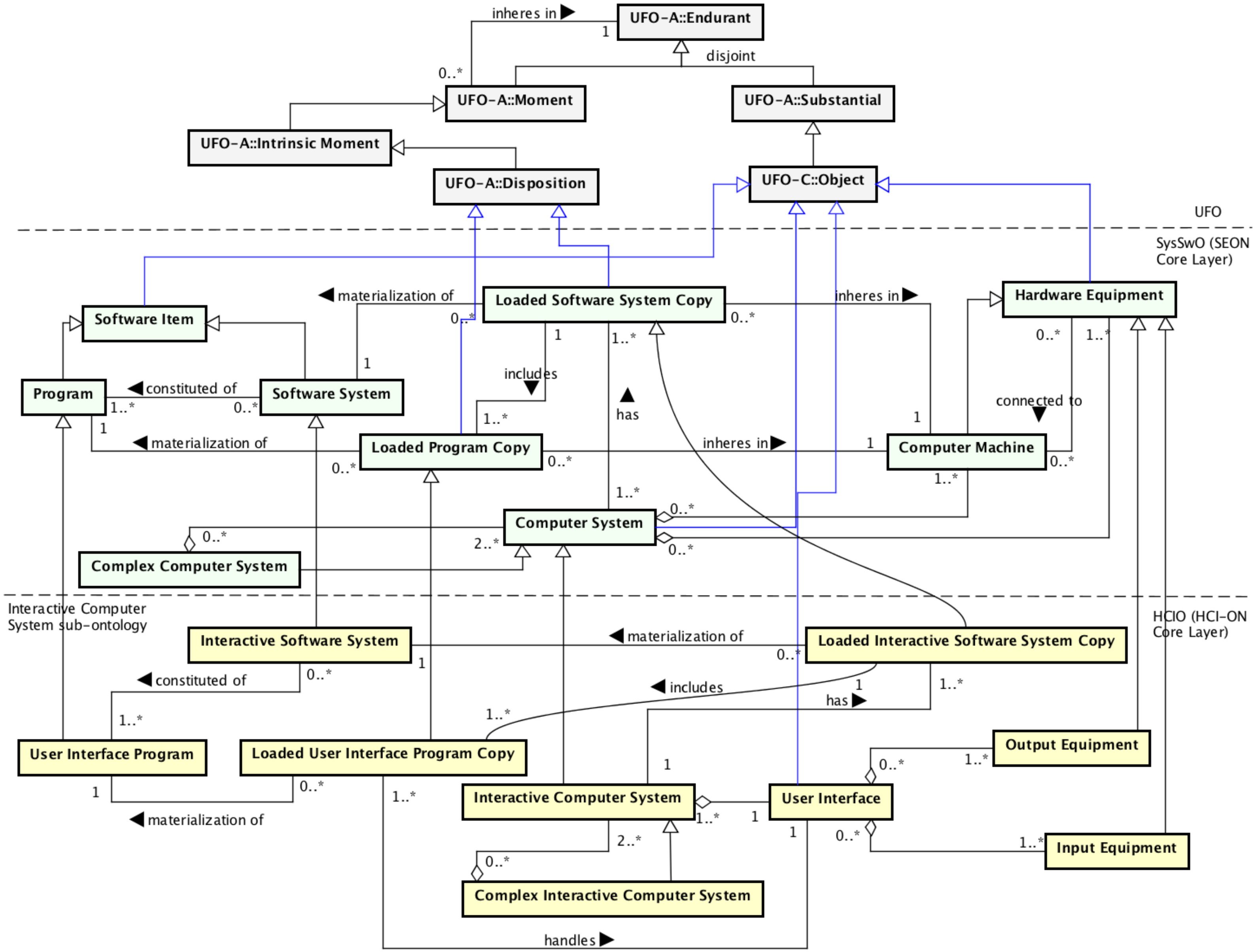


Figure 5.4 – Interactive Computer System Sub-ontology.

Computer System) composed of a set of *Computer Machines* and peripheral devices (*Hardware Equipment*) connected to them. The striking feature of an ***Interactive Computer System*** is that it has a ***User Interface***, a complex *Object* that is composed of ***Input Equipment*** and ***Output Equipment*** connected to the *Computer Machine*. If an ***Output Equipment*** is part of a ***User Interface*** of an ***Interactive Computer System***, then this ***Interactive Computer System*** should be constituted of the *Computer Machine* that connects the ***Output Equipment***. The same applies to ***Input Equipment***. These constraints are addressed by the following axioms:

A 5.1. $\forall$ *oeq*: Output Equipment, *ui*: User Interface, *ics*: Interactive Computer System partOf(*oeq*, *ui*) $\wedge$ partOf(*ui*, *ics*) $\rightarrow$ ($\exists$ *cm*: Computer Machine partOf(*cm*, *ics*) $\wedge$ connectedTo(*oeq*, *cm*))

A 5.2. $\forall$ *ieq*: Input Equipment, *ui*: User Interface, *ics*: Interactive Computer System partOf(*ieq*, *ui*) $\wedge$ partOf(*ui*, *ics*) $\rightarrow$ ($\exists$ *cm*: Computer Machine partOf(*cm*, *ics*) $\wedge$ connectedTo(*ieq*, *cm*))

When a *Hardware Equipment* is both an ***Input Equipment*** and an ***Output Equipment*** it is said an ***IO Equipment***. Thus:

A 5.3. $\forall$ *he*: Hardware Equipment Input Equipment(*he*) $\wedge$ OutputEquipment(*he*) $\rightarrow$ IOEquipment(*he*)

Regarding software, an ***Interactive Computer System*** has a set of *Software Systems* (***Interactive Software System***) loaded in its *Computer Machines* (***Loaded Interactive Software System Copies***). The programs that constitute these systems are instances of *Program*. Some of them deal with aspects related to the user interface and, thus, are instances of ***User Interface Program***. Thus, an ***Interactive Computer System*** has a ***User Interface*** and the copies of the programs loaded in its computer (***Loaded User Interface Programs Copy***) that handle its ***User Interface***. The following axiom applies:

A 5.4. $\forall$ *ui*: User Interface, *ics*: Interactive Computer System, *lissc*: Loaded Interactive Software System Copy partOf(*ui*, *ics*) $\wedge$ has(*ics*, *lissc*) $\rightarrow$ $\exists$ *luipc*: Loaded User Interface Program Copy includes(*lissc*, *luipc*) $\wedge$ handles(*luipc*, *ui*)

Interactive Computer Systems can form another ***Interactive Computer System***, which is said a ***Complex Interactive Computer System***.

In John's case (Figure 5.1), the ***Interactive Computer System*** is composed of John's desktop computer (a *Computer Machine*) and its peripheral devices (*Hardware Equipment*), Google's and Expedia's servers (*Computer Machines*) and other *Hardware Equipment* connected to them, plus the software systems loaded in those machines, such as the copies of the operating system and Chrome browser running in John's computer, as well as the copy of the Expedia's system running in one of the Expedia's servers (***Loaded Interactive Software System Copies***). Chrome, Expedia's Travel Booking System and the Operating System are instances of ***Interactive Software System***.

The ***User Interface*** is composed of, among others, the mouse, the keyboard (***Input Equipment***) and the monitor (***Output Equipment***) connected to John's desktop computer (*Computer Machine*), and has its elements handled by the copies of the programs loaded in that computer (***Loaded User Interface Programs***), such as the loaded copy of Chrome's program responsible for displaying Chrome's graphical window.

In Rino's case (Figure 5.2), the ***Interactive Computer System*** is the *Computer System* composed of Rino's Apple Watch computer (a *Computer Machine*) and its attached devices (*Hardware Equipment*), plus the software system loaded in the machine, such as the copies of the iOS, Siri and Activity (***Loaded Interactive Software System Copies***).

iOS, Siri and Activity (***Interactive Software System***) are constituted by programs and some of them handle user interface elements/widgets. The ***User Interface*** is composed of the microphone, the crown, the sensors (***Input Equipment***), the speaker, the taptic engine (***Output Equipment***) and the touch screen (***IO Equipment***), among others.

5.2.2 User Sub-Ontology

This sub-ontology focuses on actions users perform in the context of a human-computer interaction and aims to answer the competence questions CQ05 to CQ09. Figure 5.5 shows the conceptual model of the User sub-ontology.

User is the role played by a *Person* that participates in a human-computer interaction. Such participation is said a ***User Participation***, which can be either intentional (***Intentional User Participation*** – or simply ***User Action***) or unintentional (***Unintentional User Participation***).

In terms of UFO, ***User Participation*** is an *Agent Participation*. ***Intentional User Participation (User Action)*** is an *Action Contribution*, i.e., an intentional participation of a ***User***. ***Intentional User Participations (User Actions)*** are caused by *Intentions* (***User Intentions***) that inhere in the ***User***. As an *Intention*, ***User Intention*** has a *Goal* (more specifically a ***User Goal***) as its propositional content. ***User Action***, as an intentional participation, is performed by the ***User*** in order to achieve a ***User Goal***. ***User Actions*** performed by a ***User*** can only be caused by ***User Intentions*** that inhere in that ***User***. This constraint is addressed by the following axiom:

A 5.5. $\forall$ *user*: User, *ua*: Intentional User Participation (User Action), *uint*: User Intention participationOf(*ua*, *user*) $\wedge$ causedBy(*ua*, *uint*) $\rightarrow$ inheresIn(*uint*, *user*)

In John's case (Figure 5.1), he is the ***User***. He has the intention of quoting air tickets (***User Intention***) to identify the one with lowest price (***User Goal***). This intention caused him to intentionally search the Internet and fill out the required form fields (***Intentional User Participation (User Action)***).

In Rino's case (Figure 5.2), he (***User***) has the intention of monitoring his performance (***User Intention***) by monitoring heart rate, time, distance and velocity in a run (***User Goal***). Thus,

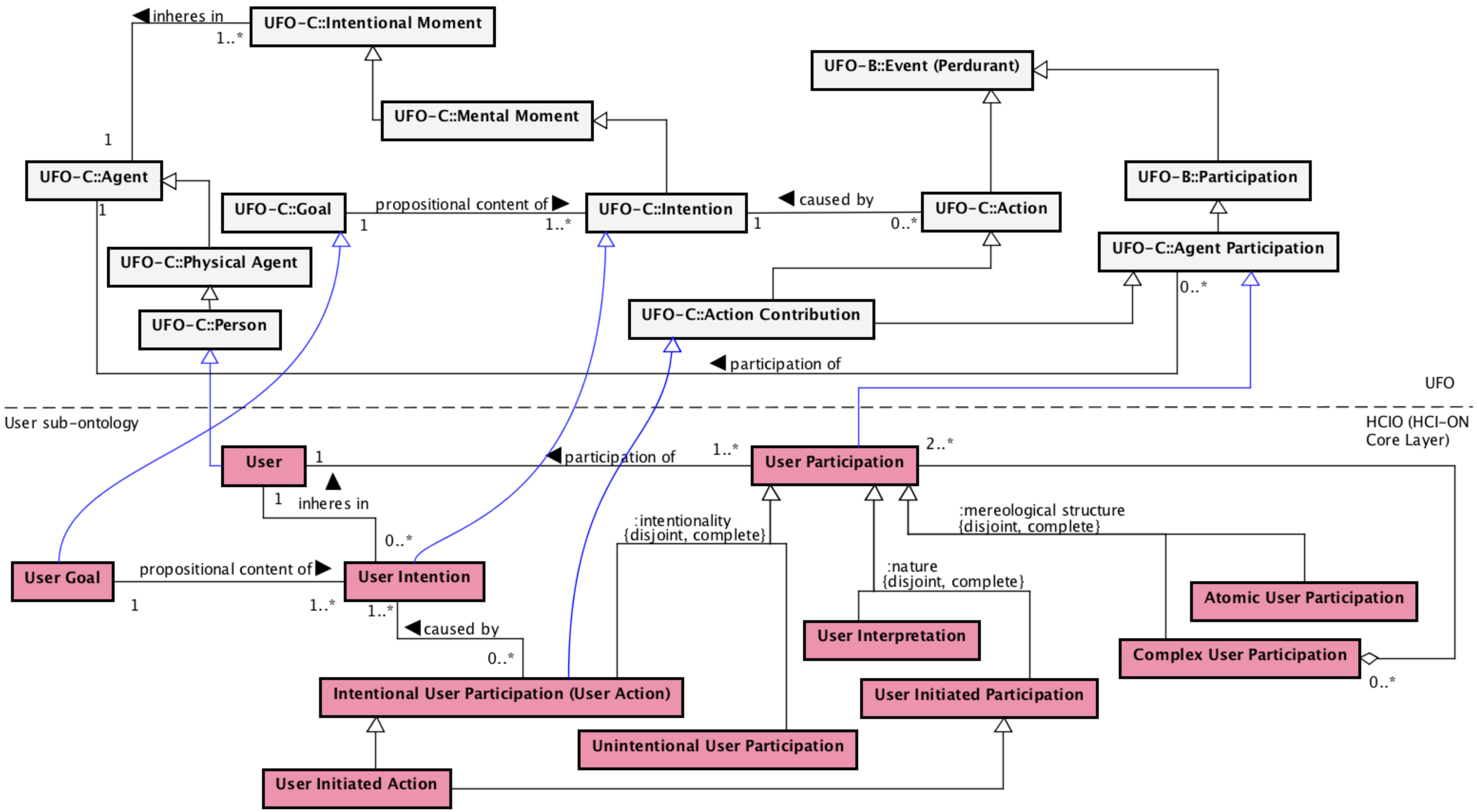


Figure 5.5 – User Sub-ontology.

he intentionally sets his smart watch to do so. For example, Rino says “Start outdoor run” to Siri (***Intentional User Participation (User Action)***). When interacting with his smart watch, Rino also acts unintentionally. For example, he unintentionally feels the watch vibrate (***Unintentional User Participation***). In addition, Rino unintentionally inputs data from his pulse (***Unintentional User Participation***).

Considering its mereological structure, a ***User Participation*** can be an ***Atomic User Participation*** or a ***Complex User Participation***, which is composed of others ***User Participations***. In a ***Complex User Participation***, all the ***User Participations*** are participations from the same ***User***. Thus:

A 5.6. $\forall$ *cup*: Complex User Participation, *user*: User, *up*: User Participation participationOf(*cup*, *user*) $\wedge$ partOf(*up*, *cup*) $\rightarrow$ participationOf(*up*, *user*)

In Rino’s case, each touch in the [+] widget to set the time for the run can be considered an ***Atomic User Participation***. Thus, the set of 30 touches composing the whole act of set 30 minutes would be a ***Complex User Participation*** composed of 30 ***Atomic User Participations*** of Rino.

In another classification, which considers the nature of participations and is orthogonal to the ones discussed above, ***User Participations*** are classified into two disjoint types: ***User Initiated Participation*** and ***User Interpretation***. ***User Initiated Participation*** refers to an act performed by the user making an input in the system. ***User Interpretation***, in turn, regards interpreting a state of the system. When a ***User Initiated Participation*** is intentional, it is said a ***User Initiated Action***.

In John’s case, he types the Expedia URL, making an input in the system. The act of typing the Expedia URL is a ***User Initiated Participation***. More than that, it is a ***User Initiated Action***, since John intentionally types the Expedia URL. When John looks at the monitor and perceives that the Expedia site was accessed, he interprets a state of the system. Thus, this act is a ***User Interpretation***. Since he was expecting to face Expedia site, his ***User Interpretation*** is also an ***Intentional User Participation***.

In Rino’s case, the act of Rino unintentionally inputting data from his heartbeat is a ***User Initiated Participation*** and its mereological intentionality is an ***Unintentional User Participation***. On the other hand, his intentional act of setting 30 minutes for the run is a ***User Initiated Action*** (i.e., an intentional ***User Initiated Participation***). When Rino feels the watch vibrate, he interprets a state of the system (***User Interpretation***). Since he was not expecting the watch to vibrate, his ***User Interpretation*** is also an ***Unintentional User Participation***.

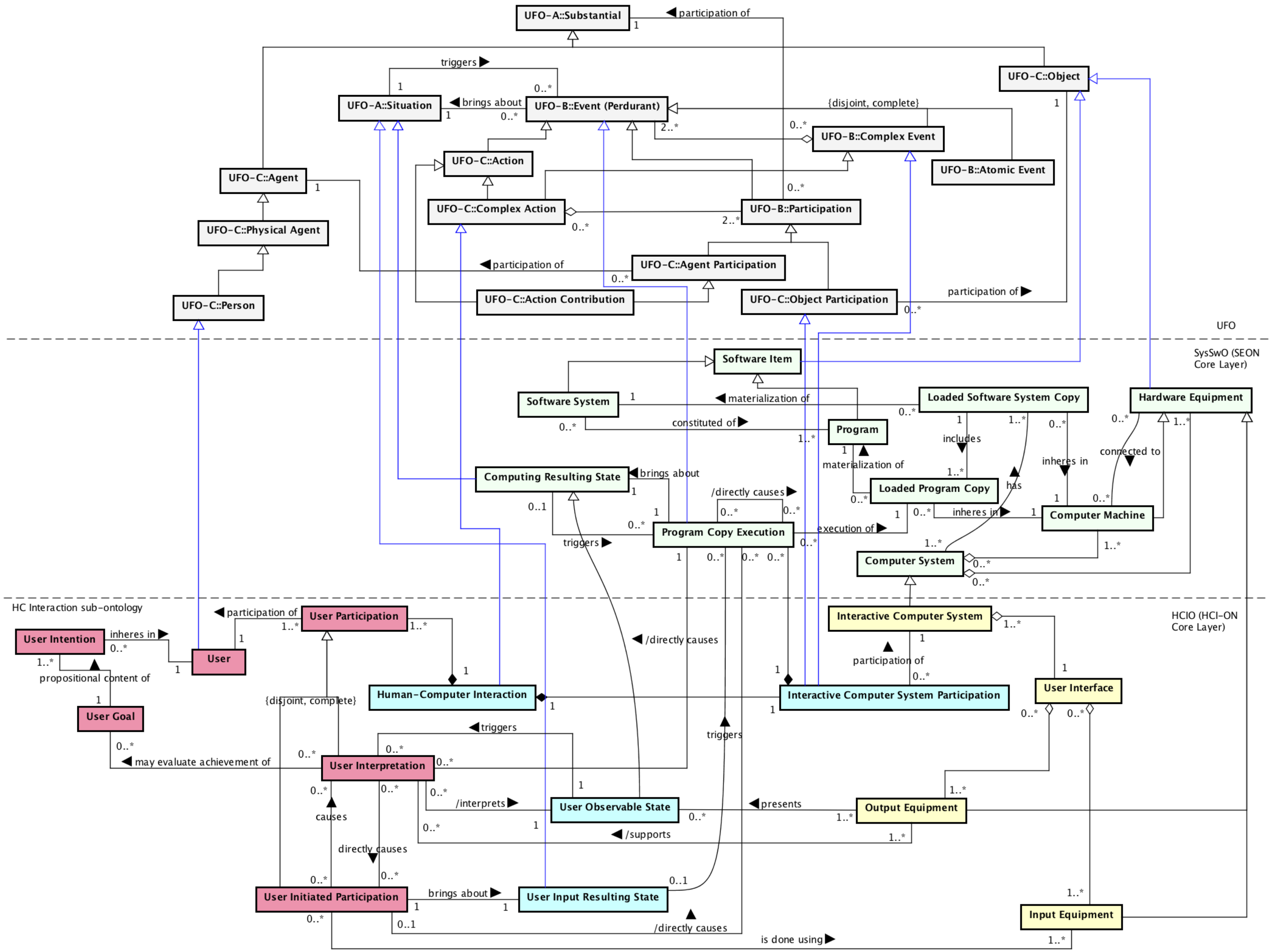


Figure 5.6 – HC Interaction Sub-ontology.

5.2.3 HC Interaction Sub-Ontology

Figure 5.6 presents the conceptual model of HC Interaction sub-ontology. This sub-ontology links concepts from the other two sub-ontologies to define what a human-computer interaction is. It aims to answer the competence questions CQ10 to CQ15.

A ***Human-Computer Interaction*** is an interaction between a ***User*** and an ***Interactive Computer System***. Thus, a ***Human-Computer Interaction*** is composed by ***User Participations*** and an ***Interactive Computer System Participation*** (which can be composed of several *Program Copy Executions*), indicating the events performed by both parties in a specific interaction. For example, when John (***User***) types the Expedia URL and the system (***Interactive Computer System***) shows the Expedia site, we have an interaction (***Human-Computer Interaction***) in which John participates by inputting data into the system and interpreting the system response (***User Participation***), and the system participates by receiving John's input, processing it, and showing Expedia site (***Interactive Computer System Participation***). It is important to notice that the events that compose a ***Human-Computer Interaction*** (i.e., the ***User Participation*** and the *Program Copy Executions* that compose the ***Interactive Computer System Participation***) can have different granularities, depending on the needs of a particular domain.

As said before (see User Sub-ontology), in terms of UFO, ***User Participation*** is an *Agent Participation*. ***Interactive Computer System Participation***, in turn, is an *Object Participation*. It is a *Complex Event*, since it aggregates all events performed by the ***Interactive Computer System*** in the context of a single ***Human-Computer Interaction***. It is an *Object Participation* because, being the ***Interactive Computer System*** an *Object*, its participation is always unintentional.

A ***User Initiated Participation*** is performed using one or more ***Input Equipment***. As a result of a ***User Initiated Participation***, a ***User Input Resulting State*** is achieved. ***User Input Resulting State*** is a *Situation* representing the data entered by the user before any program execution. This situation triggers *Program Copy Executions*. Moreover, we can say that ***User Initiated Participation*** directly causes *Program Copy Execution*. *Program Copy Execution* brings about a *Computing Resulting State* (internal computer state), which, in turn, can trigger other *Program Copy Executions*. Thus, a *Program Copy Execution* can directly cause new *Program Copy Executions*.

In John's case, using the mouse (***Input Equipment***) and keyboard (***Input Equipment***), he types the Expedia URL in the Chrome navigation bar and presses the enter key (***User Initiated Participation***). The situation resulting from the input action (***User Input Resulting State***) triggers the execution of programs that search for the Expedia site and show it (*Program Copy Execution*). In this context, the execution of the program that searches for the Expedia site produces a *Computing Resulting State* (e.g., the state in which the Expedia site is found) that causes the execution of the program that shows the Expedia site in the navigator.

Some *Program Copy Executions* can bring about a special type of *Computing Resulting State*, the one that is perceivable by user, said ***User Observable State***. A ***User Observable State***, thus, triggers ***User Interpretation***. Therefore, we say that *Program Copy Execution* directly causes ***User Interpretation***. In this context, the following axiom applies:

A 5.7. $\forall$ *ui*: User Interpretation, *uos*: User Observable State, *pce*: Program Copy Execution, *uirs*: User Input Resulting State, *uip*: User Initiated Participation triggers(*uos*, *ui*) $\wedge$ bringsAbout(*pce*, *uos*) $\wedge$ triggers(*uirs*, *pce*) $\wedge$ bringsAbout(*uip*, *uirs*) $\rightarrow$ causes(*uip*, *ui*)

It is important to say that (A5.7) does not constrain who are the users involved in the ***User Initiated Participation*** and in the ***User Interpretation***, i.e., the ***User*** in the ***User Initiated Participation*** may be different from the one involved in the ***User Interpretation***.

A ***User Observable State*** triggers ***User Interpretation*** and is presented by one or more ***Output Equipment***. Thus, ***Output Equipment*** supports ***User Interpretation***, i.e.:

A 5.8. $\forall$ *oe*: Output Equipment, *uos*: User Observable State, *ui*: User Interpretation presents(*oe*, *uos*) $\wedge$ triggers(*uos*, *ui*) $\rightarrow$ supports(*oe*, *ui*)

In John's case, the execution of programs triggered by the URL entered by him (***User Input Resulting State***) brings about the situation where the Expedia site is shown (***User Observable State***) in John's monitor (***Output Equipment***). John interprets the information showed in Expedia site (***User Interpretation***) and fills out the required fields to quote air tickets (***User Initiated Participation***).

In a ***Human-Computer Interaction***, both the ***Input*** and ***Output Equipment*** involved in the interaction should be part of the ***User Interface*** of the ***Interactive Computer System*** that participates in the interaction.

Since ***User Initiated Participation*** typically leads to some processing inside the system that results in an output to the user who can interpret it, we say that ***User Initiated Participation*** (indirectly) causes ***User Interpretation***. ***User Interpretation*** may cause the ***User*** to act again. Thus, ***User Interpretation*** may directly cause (in terms of UFO) ***User Initiated Participations***. In the example above, the act of John entering Expedia URL (***User Initiated Participation***) leads to programs execution that brings about the exhibition of Expedia site (***User Observable State***). John interprets the Expedia site (***User Interpretation***) and acts again, filling out the required fields (***User Initiated Participation***). Thus, the act of John entering Expedia URL indirect caused him to interpret Expedia site. This interpretation, in turn, directly caused him to fill out the required fields.

When a ***User Participation*** is intentional, it means that the ***User*** performed it considering some goal (***User Goal***). Interpretations performed by the user (***User Interpretation***s) may evaluate ***User Goal*** achievement. This is a case of goal-driven behavior, in which a cycle of actions (***User Participation***) are repeated until the user achieves his/her goal (***User Initiated***

Participation (indirectly) causes ***User Interpretation*** and ***User Interpretations*** (directly) causes ***User Initiated Participations***).

For example, when John types the Expedia URL (***User Initiated Participation***), he has the goal of accessing Expedia site (***User Goal***). When he sees that he has accessed the Expedia site (***User Interpretation***), he notices that he has achieved his goal in that particular participation. If we consider the whole scenario in John's case, his goal is to identify the air ticket with lowest price. With this goal in mind, he interacts with the system. He performs several actions (e.g., types the Expedia URL, fills up the form, clicks button) (***User Initiated Participation***). After each action, he perceives (***User Interpretation***) that his goal was not achieved and keeps acting until the system shows flight options and he identifies the one with lowest price, achieving his goal. In this scenario, John's input actions (***User Initiated Participation***) caused him to interpret the system outputs (***User Interpretation***). John's interpretations (***User Interpretation***), in turn, caused him to act again (***User Initiated Participation***), until he has achieved his goal.

The HC Interaction sub-ontology main purpose is to define what human-computer interaction is. In view of what was discussed, in summary, a ***Human-Computer Interaction*** is a *Complex Event* composed of ***User Participation*** and ***Interactive Computer System Participation***. ***User Participation*** can involve both ***User Initiated Participations*** and ***User Interpretations***. ***Interactive Computer System Participation*** regards the set of *Program Executions* performed by the ***Interactive Computer System*** in the interaction.

5.3 HCIO Evaluation

To evaluate HCIO, we performed Ontology Verification & Validation (V&V) activities. Considering the guidelines proposed by SABiO (Falbo, 2014), HCIO was evaluated by using two evaluation approaches: *assessment by human approach* and *data-driven approach* (Brank; Grobelnik; Mladenić, 2005). In the first, we performed a verification activity by means of expert judgment, in which we checked if the concepts, relations and axioms defined in HCIO are able to answer the competency questions. Moreover, the ontology specification was peer-reviewed by two domain experts. In the second, since a reference ontology should be able to represent real world situations, to validate HCIO, we instantiated its concepts and relations using data extracted from real cases. V&V activities were performed manually, considering the reference ontology.

To achieve the HCIO version showed in this chapter, we performed three cycles (i.e., iterated three times) on SABiO's design and evaluation activities. Each cycle resulted in a version of HCIO, which improved the previous one. After producing each version of HCIO, we evaluated it by performing V&V activities and submitting the ontology specification and the V&V results to the evaluation by domain experts. Based on the evaluation results, we improved the ontology and evaluated it again until we reached the current version. Some situations the domain experts pointed out as not properly covered by previous versions of HCIO include: a user can interact with the system without a goal in mind (e.g., the user can move his/her arm in a smart home and turn on the light without intending to do so); the same equipment can be used as input and output equipment at the same time (e.g., touch screens); more than one user can interact with the system at the same time (e.g., video games). Feedbacks like these helped us to improve the ontology. They also showed us that we needed to understand the different kinds of interaction (see Section 2.1.1) to develop an ontology able to represent different situations of the real world. Next, we present some results of V&V activities related to the current version of HCIO.

5.3.1 Verification by Experts

For verifying HCIO, we started by manually checking if the concepts, relations and axioms defined in HCIO are able to answer its competency questions (CQs). This approach enabled us to check not only if the CQs were answered, but also whether there were irrelevant elements in the ontology, i.e., elements that do not contribute to answer any of the questions. Table 5.2 illustrates this verification process for HCIO, showing which elements of HCIO (concepts, relations, properties and axioms) answer the CQs. Concerning axioms, only the ones presented in this chapter were included in the table. The table can also be used as a traceability tool, supporting ontology change management. Verification results showed that HCIO is able to answer the competency questions (i.e., the ontology addresses the established scope) and that it contains the sufficient and necessary elements to do so.

Table 5.2 – HCIO verification against its competency questions.

CQ Id	Description, Concepts and *Relations*	**Axioms**
CQ01	**What is an Interactive Computer System?** Interactive Computer System is *subtype of* Computer System that *has* User Interface and *includes* Loaded Interactive Software System Copy, which, in turn, is *subtype of* Loaded Software System Copy and is *materialization of* Interactive Software System	
CQ02	**What is an Interactive Software System?** Interactive Software System is *subtype of* Software System that *is constituted of* User Interface Loaded User Interface Program Copy is *subtype of* Loaded Program Copy that *handles* User Interface and is *materialization of* User Interface Program	A5.4
CQ03	**What is a complex interactive computer system?** Complex Interactive Computer System is *subtype of* Interactive Computer System that *is composed of* other Interactive Computer Systems	
CQ04	**What does make up the user interface of an interactive computer system**? Interactive Computer System *has* User Interface *constituted of* Output Equipment and Input Equipment, which are *subtype of* Hardware Equipment	A5.1, A5.2, A5.3
CQ05	**What is a user?** User is *subtype of* Person who performs (*participation of*) User Participation	
CQ06	**How can a user interact in human-computer interactions?** User Participation is the *participation of* a User User Initiated Participation and User Interpretation are *subtypes of* User Participation User Initiated Action is *subtype of* User Initiated Participation	
CQ07	**Considering intentionality, how can a user interact with an interactive computer system?** User Participation is the *participation of* a User Intentional User Participation (User Action) and Unintentional User Participation are *subtypes of* User Participation	

Table 5.2 – *Continued from previous page*

CQ Id	Description, Concepts and *Relations*	Axioms
	User Initiated Action is *subtype of* Intentional User Participation (User Action)	
CQ08	**Why does a user intentionally interact with an interactive computer system?** User Goal is the *propositional content of* User Intention that *inheres in* User Intentional User Participation (User Action) is *subtype of* User Participation *caused by* User Intention	A5.5
CQ09	**What does make up a complex user participation?** Atomic User Participation and Complex User Participation are *subtypes of* User Participation Complex User Participation *is composed of* other User Participation	A5.6
CQ10	**What is a human-computer interaction?** Human-Computer Interaction *is constituted of* User Participation and Interactive Computer System Participation	
CQ11	**Considering the human-computer interaction, how can a user participation cause another user participation?** User Initiated Participation *causes* User Interpretation that *directly causes* User Initiated Participation	
CQ12	**How is a user input processed by an interactive computer system?** User Initiated Participation *brings about* User Input Resulting State that *triggers* Program Copy Execution	
CQ13	**How does a user receive an output from an interactive computer system?** Output Equipment *presents* User Observable State	
CQ14	**How is a user input processed by an interactive computer system and how is the corresponding output presented to him/her?** User Initiated Participation *is done using* Input Equipment and *brings about* User Input Resulting State that *triggers* Program Copy Execution. Thus, User Initiated Participation *directly causes* Program Copy Execution Program Copy Execution *brings about* User Observable State that is *subtype of* Computing Resulting State. User Observable State *triggers* User Interpretation that *interprets* User Observable State. Thus, Program Copy Execution *directly causes* User Interpretation. Output Equipment *presents* User Observable State. Thus, Output Equipment *supports* User Interpretation	A5.7, A5.8
CQ15	**How does a user evaluate if his/her goal was achieved in a human-computer interaction?** User Interpretation *may evaluate achievement of* User Goal	

5.3.2 Validation

Concerning ontology validation, the ontology should be able to properly represent real world situations (Falbo, 2014). Based on that, we instantiated the ontology using data extracted from the John and Rino's cases. When describing HCIO conceptual models, we included some instances from these cases as examples. Next, in Table 5.3, we present a summary containing some instances extracted from Rino's case (Figure 5.2). The complete instantiation of both cases and the HCIO dictionary of terms are available at (Costa; Barcellos; Falbo, 2020a). Although in this chapter we instantiate HCIO considering only the two cases described in Section 5.1, during validation we also considered several other cases (e.g., gesture and haptic interaction, interaction with a smart house, physiological interaction, interaction with a collaborative system) to ensure that HCIO is able to represent them. The successful instantiation of HCIO with data coming from real cases gave us indications of the appropriateness of the proposed ontology as a reference model.

Table 5.3 – HCIO Instantiation - Rino's case

Interactive Computer System Sub-ontology	
Interactive Computer System	Rino's Apple Watch
Computer Machine	Apple Watch computer
User Interface	The whole Apple watch features like the crown, the touch screen, the microphone, the sensors, the speaker, the taptic engine
Input Equipment	Apple watch touch screen, microphone, crown, sensors
Output Equipment	Apple watch touch screen, speaker, taptic engine
Interactive Software System	iOS, Siri, Activity
Loaded Interactive Software System Copy	Copies of the iOS, Siri and Activity loaded in Rino's Apple Watch
User Interface Program	Programs constituting iOS, Siri and Activity that handle User Interface elements
Loaded User Interface Program Copy	Copies of the programs constituting iOS, Siri and Activity that handle User Interface elements loaded in Rino's Apple Watch
User Sub-ontology	
User	Rino
User Intention	Monitor performance
User Goal	Monitoring heart rate, time, distance and velocity in a run
User Participation (UP)	UP1 = Rino presses the Apple watch crown (Fig. 5.2, 2nd picture) UP2 = Rino sees a message from Siri on the watch screen (Fig. 5.2, 3rd picture) UP3 = Rino says "Start outdoor run" (Fig. 5.2, 4th picture) UP4 = Rino sees the Activity app opened (Fig. 5.2, 5th picture) UP5 = Rino touches the [...] widget (Fig. 5.2, 5th picture) UP6 = Rino sees a new screen from the Activity app on the watch (Fig. 5.2, 6th picture) UP7 = Rino touches the time widget (Fig. 5.2, 6th picture) UP8 = Rino sees a new screen from the Activity app on the watch (Fig. 5.2, 7th picture) UP9 = Rino touches the [+] widget 30 times (Fig. 5.2, 7th picture) UP10 = Rino touches the start widget (Fig. 5.2, 8th picture) UP11 = Rino sees the countdown on the screen (Fig. 5.2, 9th picture) UP12 = Rino turns his wrist to activate the watch screen to check his heart rate and performance (Fig. 5.2, 10th picture) UP13 = Rino sees that his heart rate is 169 BPM, distance 0,91 MI and pace 4'66"/MI (Fig. 5.2, 10th picture) UP14 = Rino feels the watch vibrate (Fig. 5.2, 11th picture) UP15 = Rino turns his wrist to activate the watch screen (Fig. 5.2, 11th picture) UP16 = Rino sees the message informing that he has reached half the way (Fig. 5.2, 11th picture) UP17 = Rino feels the watch vibrate (Fig. 5.2, 12th picture) UP18 = Rino turns his wrist to activate the watch screen (Fig. 5.2, 12th picture) UP19 = Rino sees the message informing that he has reached his goal (Fig. 5.2, 12th picture)
User Initiated Action	UP1, UP3, UP5, UP7, UP9, UP10, UP12, UP15, UP18
User Interpretation	UP2, UP4, UP6, UP8, UP11, UP13, UP14, UP16, UP17, UP19
Complex User Participation	UP9 (considering each touch to reach 30 minutes as an Atomic User Participation)
Unintentional User Participation	UP14, UP17
HC Interaction Sub-ontology	
For simplification reasons, in this sub-ontology we *instantiate* only *one* human-computer interaction. *The others involved in the Rino's case are similar to the one presented in the following*	
Interactive Computer System Participation (ICSP)	ICSP1 = The set of program executions and other events involving the Apple watch's computer system when interacting with Rino in the context of UP1 + UP2
Human-Computer Interaction	The interaction constituted of the Complex User Participation UP1 + UP2 and the Interactive Computer System Participation ICSP1
User Input Resulting State (UIRS)	UIRS1 = The situation achieved as a result of performing UP1 (i.e., the Apple watch crown pressed)

Table 5.3 – *Continued from previous page*

User Observable State (UOS)	UOS1 = The situation observable in the Apple watch screen in UP2 (i.e., the Siri message shown in the screen)
Program Copy Execution	The set of executions of program copies constituting the iOS, Siri and Activity that are loaded in the Rino's Apple watch, which led from UIRS1 to UOS1

After HCIO evaluation, we implemented HCIO's operational version using the open-source editor Protégé 5.5.0[2], which supports the construction of OWL[3] models, and its in-built reasoner HermiT 1.4.3. HCIO machine-readable version is available at <http://bit.ly/hcioOWL>. To provide a graphical visualization of the operational ontology, we used WebVOWL[4] to create a web-based visualization, which is available at <http://bit.ly/hcioWebVOWL>. It also can be visualized with OWL-GrEd[5] (UML style graphical editor for OWL) at <http://bit.ly/hcioGrEd> and Turtle[6] (textual syntax for RDF) at <http://bit.ly/hcioTurtle>.

5.4 Discussion

Concerning different interaction types (see Section 2.1.1), HCIO is able to represent both *explicit* and *implicit* interaction (what we call respectively as intentional and unintentional). For example, in John's case, when he performs intentional actions driven by his goal of quoting flights costs, we have explicit interaction. We have implicit interaction when Rino unintentionally inputs data from his pulse rate. HCIO is also able to represent *goal* and *data* or *event-driven behavior*. As previously discussed, in John and Rino's cases, they interact with the computer system driven by the goals they want to achieve, thus, we have goal-driven behavior. However, when Rino perceives the watch vibrate, we have event-driven behavior.

To represent different types of user actions, we decided to use User Participation as a general term based on UFO. From this umbrella concept, there are subtypes of user participation that allow representing user intentional or unintentional actions (related to user input) and interpretation (related to system output). Regarding unintentional participation, it represents unintentional user actions mostly because it has no associated goal. It also represents unconscious and uncontrolled actions related to human vital activities such as physiological functions, that are indeed unintentional.

Since HCIO is devoted to human-computer interaction, it does not represent interaction between humans and non-computer systems (e.g., a typewriter or a shower). HCIO allows representing traditional interactive computer systems, computer household appliances (e.g., a coffee machine that has software and hardware able to run programs) and even more complex systems such as IoT (Internet of Things), among others. As for actions, all user actions considered

2 Available for download at <https://protege.stanford.edu/products.php#desktop-protege>
3 W3C Web ontology Language, <https://www.w3.org/OWL/>
4 Web-based Visualization of Ontologies, <http://vowl.visualdataweb.org/webvowl.html>
5 OWLGrEd online visualization available at <http://owlgred.lumii.lv/online_visualization>
6 W3C Turtle, <https://www.w3.org/TR/turtle/>

in HCIO are actions the user performs to interact with the computer system. Thus, actions that do not involve interaction with a computer system (e.g., if the user drinks water when typing a URL in his/her computer) are not covered.

Although in this chapter we have explored examples of HCI involving a single user, HCIO allows representing situations common in collaborative systems, where different users interact with the same interactive computer system at the same time. For these cases, the ontology represents the interaction from the point of view of each individual user. The common element in this collaborative interaction is the interactive computer system. For example, an interactive computer system allows a certain user to initiate an action (e.g., a user types a text in a shared document) and another user to interpret the system response to that action (e.g., another user sees the document and interprets the text added by the first user).

In HCIO, we consider that the user observes situations resulting from the execution of events (e.g., a message shown in the screen after the execution of programs). However, when interacting with an interactive computer system, a user could also observe the execution of the event itself (e.g., the execution of a video on YouTube). Currently, this is not addressed in HCIO (it can be explored in future work).

In the context of an interaction, HCIO allows representing situations in which there is a disruption to the communication process due to failure or error in the computer system. This may occur when a user input (User Input Resulting State) does not lead to the execution of programs or the execution of programs does not lead to new system output (User Observable State). For example, the user clicks a button, but nothing happens because there is no program associated with that button (an implementation error) or due to a failure in the corresponding program. This can be particularly useful to represent interactions with prototypes and support prototype evaluation. In HCIO, User Observable State is related to what Nielsen (1993) calls user feedback. User feedback refers to a basic characteristic of usable interfaces in which the system should continuously inform the user about what it is doing (Nielsen, 1993). When the system fails (e.g., because it was not properly designed or due to hardware malfunction), it may not provide new feedback for the user. For example, if after the user provides an input to the system it behaves as if nothing had happened, the absence of change (e.g., because the program execution failed) is itself a User Observable State, which will be interpreted by the user, who will conclude that something went wrong. HCIO does not represent what went wrong (e.g., if the failure was caused by a design problem – the program was incorrect – or a hardware problem). It does not explicitly address the problems because we consider that this issue (hardware and software malfunction) is more related to Software Engineering and, thus, should not be treated as a core aspect of the HCI phenomenon.

Being a core reference ontology, HCIO aims to describe core aspects of the HCI phenomenon. Thus, particularities of specific types of interaction (e.g., gesture or haptic interaction) are not addressed. Therefore, forms of input control (e.g., by hand, by blinking eye, brain activity, by moving arms, whole-body), input (e.g., direct manipulation, pointing, mouse) and output (e.g., visual, graph, screen) modalities, types of user interface (e.g., haptic-tactile interface, brain-computer interface), types of interactive computer systems (e.g., intelligent house, physiological computing, adaptive systems) are outside of HCIO scope. Ontologies focusing on particular types of interactions and involved elements should be defined by reusing/specializing HCIO concepts. For example, if one wants to create a brain-computer interface ontology and needs to represent brain activities, he/she can extend the User concept (i.e., create new concepts related to it) to address user body and brain (and even brain areas), and can extend the User Initiated Participation or User Initiated Action (intentional brain activity) concepts to represent brainwave and relate it to brain areas in order to represent which area of the brain is responsible for inputs that generate different behaviors in the system.

As a core ontology, HCIO should allow representing situations under the perspective of the different interaction paradigms presented in Section 2.1.1. On one hand, HCIO is general enough to address the different paradigms at a higher abstraction level. On the other hand, by specializing HCIO concepts, it is possible to develop specific domain ontologies focusing on each paradigm. For example, by specializing HCIO, we have developed an ontology addressing the HCI phenomenon under the Cognitive Science perspective (Costa; Barcellos; Falbo, 2020b).

5.5 Comparing HCIO to other Ontologies

Considering the ontologies we found in our secondary study (see Chapter 3), eleven ontologies (#01, #05, #06, #07, #09, #11, #12, #13, #14, #15 and #18) cover aspects of the HCI phenomenon. When analyzing these ontologies, we noticed that #05, #06, #07, #11 and #18 address the HCI system part of the interaction focusing only on web UI aspects; #12 and #13 address the HCI user part by means of Persona. #09 and #14 address both user and system, but focusing specifically on gesture and haptic interaction, respectively. Therefore, we consider #01 and #15 the ontologies most related to HCIO. Concerning #01, it is the oldest ontology addressing the HCI phenomenon we found. It has a good coverage of this phenomenon, centered in four main notions: Participant, Interaction, Purpose and Interface. Its top-level statement of the relationships between these notions is: “Any interaction takes place through one or more interfaces and involves two or more participants who each has one or more purposes for the interaction”. Based on this statement, (Storrs, 1994) clarifies the four notions. Regarding Participants, he considers People, Social Groups and Computers, as well as Participant Roles (Owner, User and Representative). Concerning Interaction, he considers several interaction characteristics (synchronous/asynchronous, directed/mediated, cooperative/individual, cheap/expensive) and Elements of Interaction (including Utterance, Dialogue and Interaction). However, the ontology is presented only in a textual form, in natural language. Concepts can be easily identified, but relations are often difficult to capture. In sum, this ontology lacks formality. HCIO, on the other hand, goes deeper in the computer side, describing what an interactive computer system is and how it participates in an interaction. Moreover, HCIO talks about types of user actions and how they are linked to events processed by machines. HCIO is grounded in a foundational ontology (UFO) and is modular (divided in three sub-ontologies). HCIO is presented as a conceptual model with axioms added to capture important constraints that are not possible to be captured by the graphical model.

Regarding #15, the ontology is divided in two modules. The Window Icon Menu Pointer User Interface ontology model (WIMP-UI) aims to capture the semantic meaning of user interface that is used by users of an application software in an interaction. The focus is on the graphical elements that compose a WIMP Interface, namely Window, Icon, Menu and Pointer, and smaller elements that compose Windows and Menus, such as Text Field, List Box, Table and so on. The User Interaction (USI) ontology model aims to represent the interaction between user and software, and includes concepts such as Interaction, User, Software, Interface, Action and Response. Comparing to HCIO, #15 deals with graphical UI elements that are not explicit in HCIO. On the other hand, it says nothing about how software produces responses, neither about what is an interactive computer system and how hardware and software are connected to it.

The main distinguishing feature of our ontology when contrasted to the other HCI ontologies is that it is a core reference ontology and has been developed taking characteristics of "beautiful ontologies" (D'Aquin; Gangemi, 2011) into account. In summary: (i) HCIO covers core aspects regarding the interaction phenomenon, providing explanation about the interaction itself and the involved parts; (ii) HCIO is a modular ontology, which favors understanding and reuse; (iii) HCIO is represented in a good level of formalism by means of conceptual models, axioms and textual descriptions; (iv) HCIO is a well-founded ontology grounded in UFO; and, finally, (v) HCIO was developed and evaluated by following SABiO method (Falbo, 2014), a well-established method used in several ontology development efforts.

5.6 Final Considerations of the Chapter

This chapter presented HCIO, a core reference ontology about the human-computer interaction phenomenon. HCIO scope was defined by means of competency questions as suggested by the adopted ontology engineering method. HCIO is grounded in UFO, reuses and extends concepts from SysSwO and is modularized in three sub-ontologies providing explanation about: (i) what an interactive computer system is, its components and its user interface; (ii) user actions taken in the course of an HCI and the user motivation to start an interaction; and (iii) how the an HCI happens. Techniques of ontology verification and validation were applied to evaluate HCIO. HCIO contributes to the HCI area by providing a common conceptualization about core aspects of the HCI phenomenon.

In the secondary study presented in Chapter 3, we found two ontologies addressing HCI phenomenon aspects. We noticed that these ontologies have been developed in isolation, to solve specific problems and, thus, there are inconsistencies among the ontologies conceptualization. When an ontology is developed for a specific application, a specific solution is built and the ontology conceptualization is biased in that application. Thus, the conceptualization is solution-dependent, not concerned to describe the domain in the reality and has little potential to reuse. A reference ontology, in turn, is a solution-independent specification and, as such, provides a reference conceptualization that can be used in several solutions. Moreover, because it provides conceptualization true to reality, it can be used for the purposes of communication and learning (Guizzardi, 2007). A core ontology, in turn, represents knowledge that spans different (sub)domains. It favors knowledge reuse and, by serving as a basis to the development of other ontologies, it promotes knowledge growth and consistency (Scherp et al., 2011). Therefore, while application ontologies are devoted to a specific application, core ontologies provide central knowledge that can be reused by several specific (domain) ontologies that, in turn, can support ontology-based solutions. Being a reference and core ontology, HCIO is able to provide benefits from these two types of ontologies.

6 HCI Evaluation Ontology - HCIEO

This chapter presents the Human-Computer Interaction Evaluation Ontology (HCIEO), a *reference ontology* that aims to provide a common conceptualization of HCI Evaluation and is a domain ontology of HCI-ON. As part of HCI-ON – the artifact proposed in this research – HCIEO is related to the Design cycle (Figure 1.1). It was developed aiming to satisfy R4 (cover HCI subdomains). As HCI-ON, it is also related to the Rigor cycle, since it contributes to grow the existing knowledge base. Section 6.1 introduces the chapter and presents two cases of HCI evaluation that are later used to demonstrate how HCIEO is able to represent real-world situations. Section 6.2 presents HCIEO, its architecture and competency questions. Section 6.3 discusses how we evaluated HCIEO. Section 6.4 makes some discussions about HCIEO conceptualization. Section 6.5 compares HCIEO and existing ontologies addressing HCI evaluation. Last, Section 6.6 closes the chapter.

6.1 Introduction

As argued in the previous chapters, in an ontology network, domain ontologies provide concepts that address subdomains of a large domain, helping grow the network and solve problems related to the subdomains. In this sense, to deal with aspects related to HCI evaluation, we developed HCIEO. Among the 35 ontologies found in our secondary study, five address aspects related to HCI evaluation (#16, #20, #22, #28 and #32), but they are not devoted to HCI evaluation (they only contain some HCI evaluation concepts). Thus, we decided to develop the Human-Computer Interaction Evaluation Ontology (HCIEO), to properly represent HCI evaluation main aspects, serving as a reference about the domain and increasing domain concepts in HCI-ON.

Before presenting HCIEO, next, we describe two scenarios of HCI evaluation, which are used later to exemplify (i.e., instantiate) HCIEO concepts. The cases were performed following reality and here they are presented by means of storyboards using fictitious names. In the first case, Rita is responsible for conducting an inspection-based evaluation of a login web user interface (UI). In the second case, Rita evaluates the same UI, but now adopting a user-based evaluation, in which a user (Alex) of the system participates.

Figure 6.1 illustrates the case where Rita evaluates the login web UI (Figure 6.1, 3rd picture) using an inspection-based evaluation method. Rita is a usability specialist and works at the usability lab of a university (1st picture). Rita is responsible for evaluating the login interface of the web system used by the students (2nd picture), in order to verify the existence of usability problems in the web site login UI (3rd picture). To carry out the evaluation, Rita adopts Nielsen's usability heuristic (Nielsen, 1993). Therefore, Rita uses a checklist containing Nielsen's 10 usability heuristics (4th picture), for example: #1: Visibility of system status and #8:

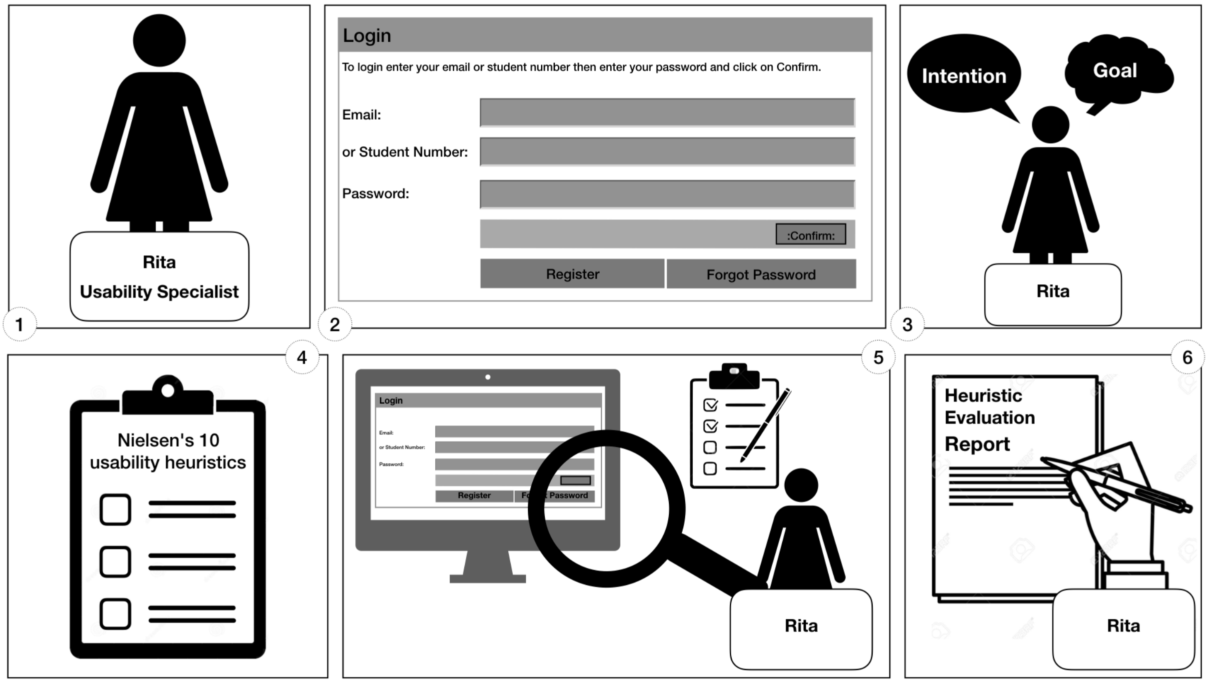


Figure 6.1 – Storyboard: Rita evaluating the UI using the Heuristic Evaluation method.

Aesthetic and minimalist design. The checklist specifies the evaluation criteria (according to the Nielsen's heuristics), to be used to evaluate the usability of the UI (4^{th} picture). Rita performs the evaluation using the heuristic evaluation method (Nielsen, 1994b), not involving the participation of users. Thus, she inspects the UI using the checklist (5^{th} picture). After the inspection, Rita prepares a report, recording the evaluation results according to the used evaluation criteria (6^{th} picture).

To complement the evaluation carried out through inspection (Figure 6.1), Rita performs a new evaluation, now with the participation of users of the system (Figure 6.2).

Consider the participation of Alex, a student that uses the system (Figure 6.2, 1^{st} picture). In this case, Rita adopts the Usability Testing (Nielsen, 1993; Rubin; Chisnell; Spool, 2008) as evaluation method, which allows observing Alex to interact with the system and, at the same time, making measurements to evaluate usability. Hence, Rita informs to Alex that he must perform the password recovery task when interacting with the UI (2^{nd} picture). Rita defines that the usability will be evaluated considering: (i) the time spent to recover a password in the first time the user used the system versus (ii) in the second time; (iii) the number of wrong clicks or touches; and (iv) the number of requests for help. It is expected that: the time to perform the password recovery task does not exceed two minutes on the first attempt to use the system and one minute on second one, there is no requests for help and no more than two wrong clicks or touches are given. To perform the evaluation, Rita uses a table to record the values measured during the evaluation (3^{rd} picture), so that they can then be compared with the corresponding expected values. Alex interacts with the system (to recover the password) (4^{th} picture). Meanwhile, Rita observes him interacting, performs measurements and records the measured values in the table (5^{th} picture).

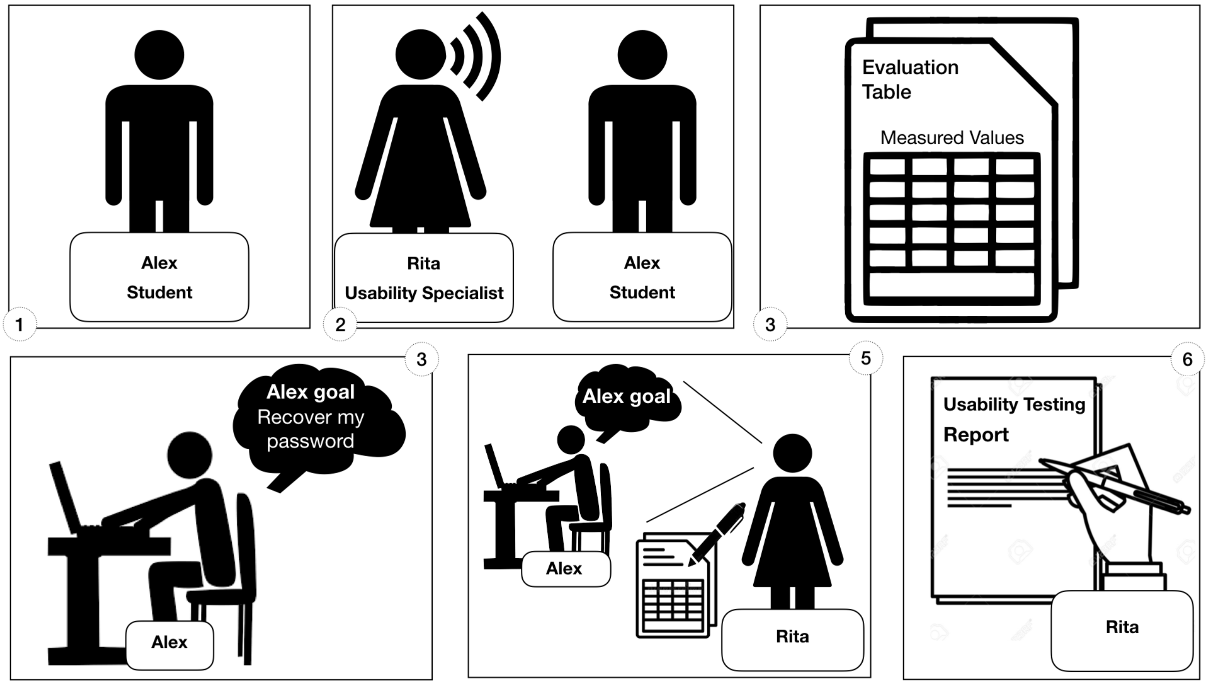


Figure 6.2 – Storyboard: Rita evaluating the UI using the Usability Testing method.

Alex asked for help three times, gave two wrong clicks and took three minutes to recover the password on the first use and a minute and a half on the second use. After the usability test, Rita elaborates a report containing the evaluation results, which considers the measured values and whether they are in accordance with the expected values (6th picture).

6.2 HCIEO

The purpose of the HCI Evaluation Ontology (HCIEO) is to establish an explicit and shared conceptualization of HCI evaluation, describing the main concepts involved in this context. The knowledge sources used to build HCIEO include standards, such as (ISO, 2019a; ISO, 2019b; ISO, 2018; ISO/IEC, 2016; ISO/IEC, 2011; ISO/IEC TR, 2010; ISO/TS, 2010; ISO/TR, 2002; ISO/IEC, 2001; ISO/IEC, 1998) and relevant literature in the HCI area, such as (Dix et al., 2004; Benyon, 2010; Rogers; Sharp; Preece, 2011; Saffer, 2010; Nielsen, 1993; Nielsen, 1994b; Rubin; Chisnell; Spool, 2008), among others. HCIEO also includes knowledge obtained from the study of the SLR ontologies. Like in HCIO development, we did not reuse the ontologies themselves because, as we explained in the Chapter 3, they have several limitations.

HCIEO is grounded in UFO (Guizzardi, 2005) and reuses concepts from six ontologies: HCIO (Human-Computer Interaction Ontology), the core ontology of HCI-ON (Chapter 5), HCIDO (Human-Computer Interaction Design Ontology) (Costa et al., 2020; Castro, 2021), a domain ontology of HCI-ON that deals with HCI design and its relation with user requirements, System Software Ontology (SysSwO) (Bringuente; Falbo; Guizzardi, 2011; Duarte et al., 2018) (Section 2.3.1.1), Software Process Ontology (SPO) (Bringuente; Falbo; Guizzardi,

2011) (Section 2.3.1.2, Core Ontology on Measurement (COM) (Barcellos; Falbo; Frauches, 2014) (Section 2.3.1.3), all of them are core ontologies of SEON, and Reference Software Requirements Ontology (RSRO) (Ruy et al., 2016) (Section 2.3.1.4), a domain ontology of SEON. HCIEO covers relevant aspects of the HCI evaluation, such as the main involved artifacts, considered evaluation criteria, evaluated characteristics, involved agents and measures that can be used to evaluate an interactive computer system. Figure 6.3 shows HCIEO and its relations with HCI-ON (HCIO and HCIDO), SEON (SysSwO, SPO, COM and RSRO) and UFO. In the figure, dependency relations mean that the source ontology reuses concepts from the target ontology.

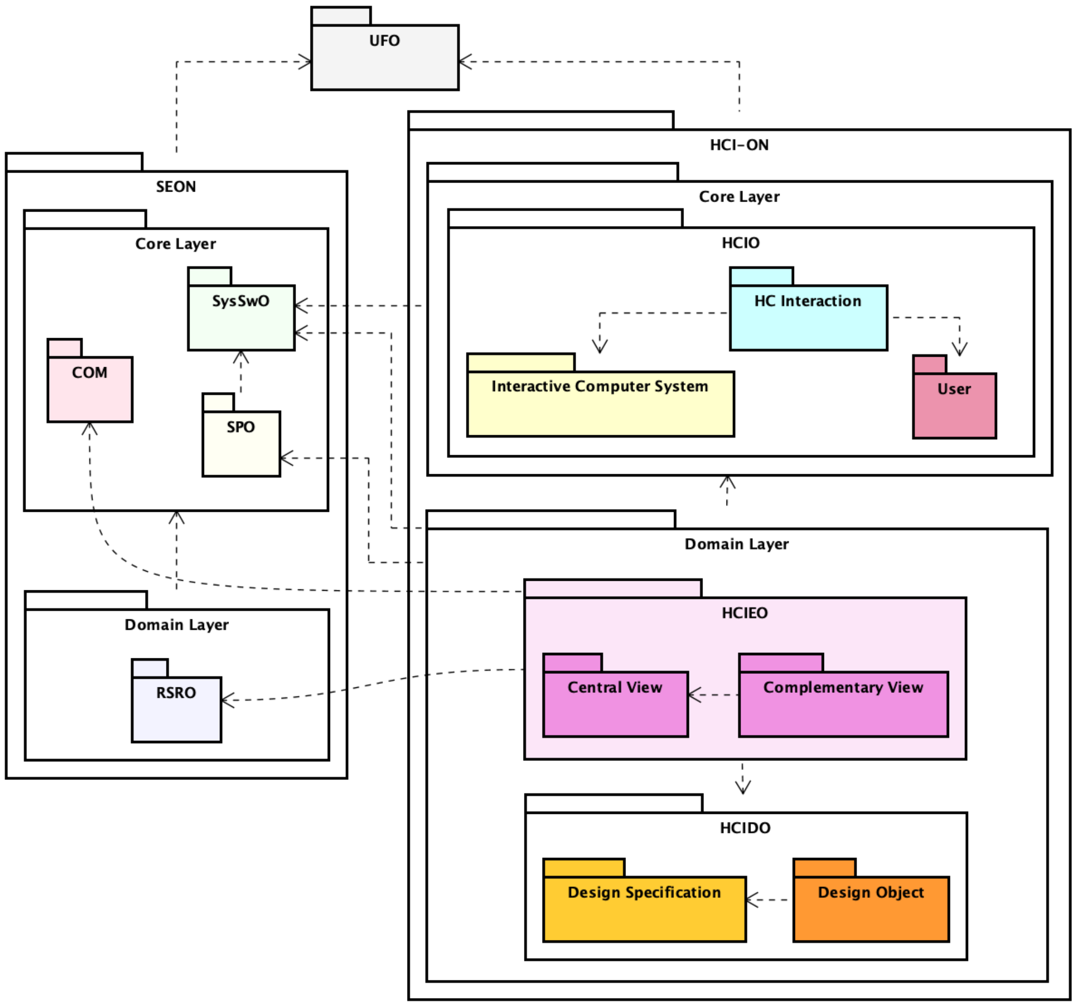


Figure 6.3 – HCIEO architecture.

As HCIO (Chapter 5), HCIEO was developed by following SABiO (Falbo, 2014). Hence, the ontology scope was defined by means of competency questions. Table 6.1 shows HCIEO competency questions (CQ). CQ01 to CQ03 help understand the motivation behind an HCI evaluation and what it evaluates. CQ04 and CQ05 are to know the criteria and artifacts used to conduct an evaluation. CQ06 and CQ07 refer to the evaluation results. CQ08 and CQ09 concern the stakeholders involved in an evaluation. CQ10 concerns the evaluation method. Finally, CQ11 and QC12 are to understand how quality characteristics are quantified in an evaluation.

Table 6.1 – HCIO Competency Questions.

Id	Description
CQ01	What leads to an HCI evaluation?
CQ02	What does an HCI evaluate?
CQ03	What quality characteristics are evaluated in an HCI evaluation?
CQ04	What criteria are applied in an HCI evaluation?
CQ05	What artifacts are used to perform an HCI evaluation?
CQ06	What is the result of an HCI evaluation?
CQ07	What does the result of an HCI evaluation report?
CQ08	Who performs an HCI evaluation?
CQ09	Who participates in an HCI evaluation?
CQ10	What method is used in an HCI evaluation?
CQ11	How can quality characteristics be quantified?
CQ12	Which are the values assigned to quality characteristics of an interactive computer system in an HCI evaluation?

In this chapter we split the HCIEO conceptual model into two aiming at a better visualization. In the conceptual models, we kept the colors used in Figure 6.3 to identify the concept source. In the models description we refer to the scenarios of use presented in Section 6.1 to exemplify HCIEO concepts. Figure 6.4 presents the central view of HCI evaluation. In the figure, doted lines separate the ontologies in layers. At the top, we have UFO, providing the general foundation. At the center, there are: SPO (SEON Core Layer), containing core concepts related to stakeholder and procedure; SysSwO (SEON Core Layer), embracing core concepts related to software artifacts; RSRO (SEON Domain Layer) and HCIDO (HCI-ON Domain Layer) encompassing concepts about requirements; and, last, HCIO (HCI-ON Core Layer), containing core concepts related to Human-Computer Interaction. At the bottom, there is HCIEO concepts, representing the central view, grounded in UFO or specialized from SPO, SysSwO, RSRO and HCIO (which are also grounded in UFO). Relations that ground concepts in UFO (i.e., specializations) are shown in blue. In addition, some relationships between concepts already presented in previous conceptual models have been omitted for a cleaner visualization of the conceptual model. Such relationships can be visualized in conceptual models presented in Chapters 2, 4 and 5.

In the text, we use *italics* to refer to UFO concepts and **bold** to HCIEO concepts. To the SEON ontologies concepts we used sans serif font family. We also used `teletype font family` to refer to HCI-ON ontologies, adopting: *`italics`* to HCIDO concepts and <u>*`underline italics`*</u> to HCIO concepts. In the models description, we present some of the axioms defined to address constraints not captured in the models.

HCI Evaluation is a *Complex Action* and, like that, it is an intentional event. It is

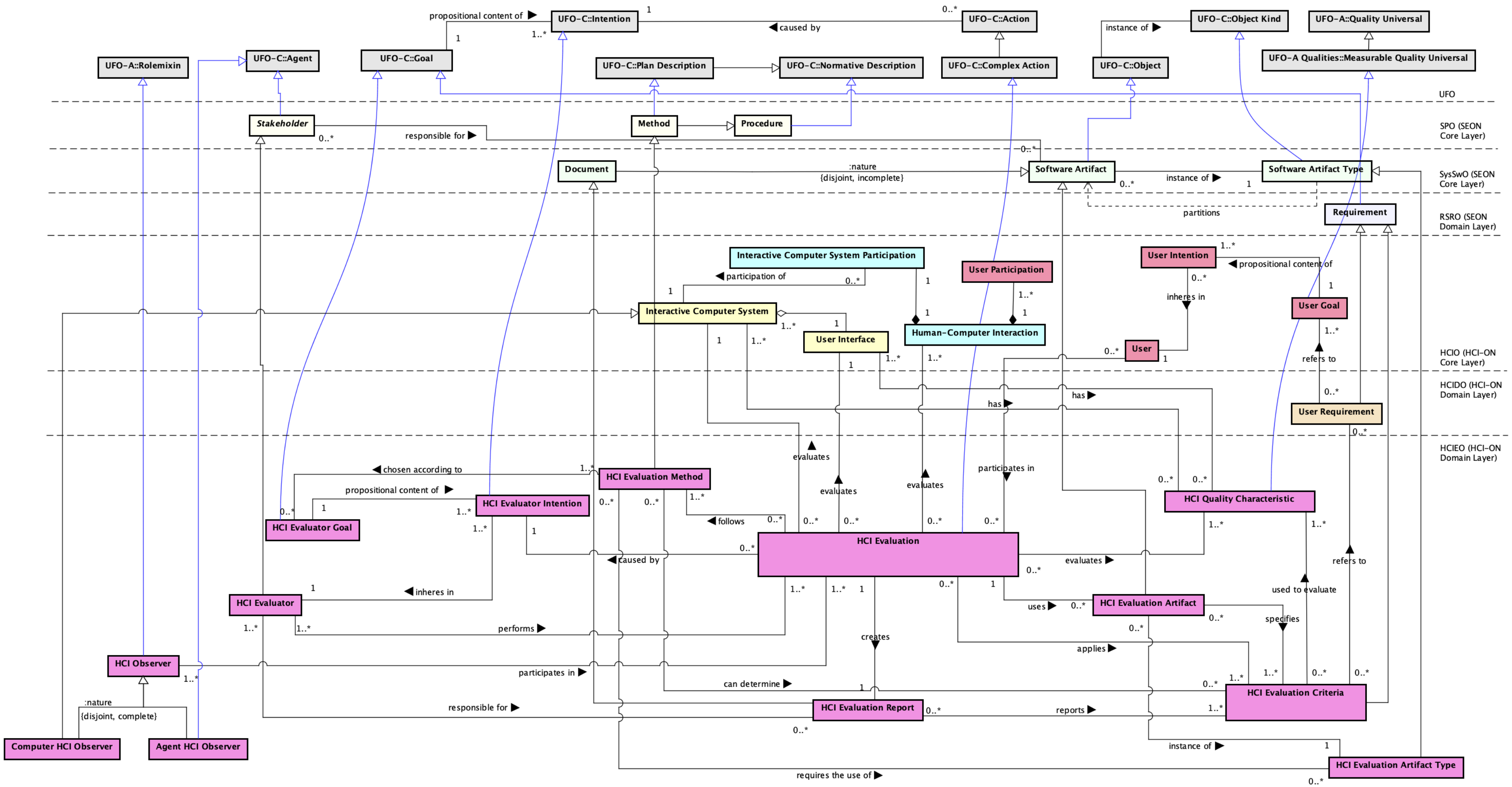


Figure 6.4 – HCIEO central view.

caused by the **HCI Evaluator Intention** that inheres in the **HCI Evaluator**. Being an *Intention*, **HCI Evaluator Intention** has a **HCI Evaluator Goal** (*Goal*) as propositional content. **HCI Evaluator** is the role played by an individual or an organization (Stakeholder) that performs the **HCI Evaluation**. In the inspection-based case (Figure 6.1) and user-based case (Figure 6.2), Rita (**HCI Evaluator**) intends to find usability problems in the web site login UI (**HCI Evaluator Intention**) motivated by making the student's web site login UI more usable and understandable (**HCI Evaluator Goal**).

HCI Observer participates in **HCI Evaluations** and is responsible for getting data through observation. **HCI Observer** is a *Rolemixin*, thus it aggregates role types whose instances have different identity principles (nature). Therefore, when an agent (*Agent*) plays the **HCI Observer** role, it is an **Agent HCI Observer**. When observation is made by an *Interactive Computer System* (e.g., by capturing interaction data and storing it an interaction log file), the **HCI Observer** is a **Computer HCI Observer**. In the cases presented above, Rita plays the role of both, **HCI Evaluator** and **HCI Observer**.

An **HCI Evaluation** consists in the systematic determination of the extent to which an *Interactive Computer System*'s (or its *User Interface*) quality characteristics (**HCI Quality Characteristic**) meet the **HCI Evaluation Criteria** applied in the evaluation. **HCI Evaluation Criteria**, in turn, are conditions, or capacity needed (Requirement), used to evaluate *Interactive Computer System*'s **HCI Quality Characteristics** and may be related to *User Requirements*.

HCI Quality Characteristics are qualities (*Quality Universal*) of an *Interactive Computer System* (manifested in its software or hardware constituents). Therefore, **HCI Evaluation** evaluates **HCI Quality Characteristics** of an *Interactive Computer System* or, more specifically, of its *User Interface*. For example, in the scenarios of use presented in Section 6.1, the HCI evaluation aimed to evaluate the usability of the interactive computer system (particularly of the web login UI) used by the students. In the first case, the HCI evaluation criteria were established through Nielsen's heuristics. In the second, Rita defined the criteria, such as the user should not need any help to use the system. Thus, the following axioms apply:

A 6.1. $\forall$ *hcie*: HCI Evaluation, *ics*: Interactive Computer System, *hciqc*: HCI Quality Characteristic evaluates(*hcie*, *ics*) $\land$ evaluates(*hcie*, *hciqc*) $\rightarrow$ has(*ics*, *hciqc*)

A 6.2. $\forall$ *hcie*: HCI Evaluation, *ui*: User Interface, *hciqc*: HCI Quality Characteristic evaluates(*hcie*, *ui*) $\land$ evaluates(*hcie*, *hciqc*) $\rightarrow$ has(*ui*, *hciqc*)

An **HCI Evaluation** follows an **HCI Evaluation Method** (e.g., Heuristic Method, Usability Testing), which is a Method that describes the actions to be performed by the **HCI Evaluator** in an **HCI Evaluation**. An **HCI Evaluation Method** is chosen according to the **HCI Evaluator Goal** and can determine the **HCI Evaluation Criteria** to be applied in an **HCI Evaluation**. Moreover, an **HCI Evaluation Method** may require the use of **HCI Evaluation**

Artifact Types (e.g., HCI evaluation checklist). Thus, to perform an **HCI Evaluation** it may be necessary to use **HCI Evaluation Artifacts** (Software Artifact), which are objects intentionally built to specify, among others, the **HCI Evaluation Criteria** to be applied in the evaluation (e.g., a checklist specifying the criteria to be applied). The **HCI Evaluation Artifacts** used in an **HCI Evaluation** should be instances of the **HCI Evaluation Artifact Types** required by the **HCI Evaluation Method** used in that **HCI Evaluation**. For example, if an HCI evaluation used an HCI evaluation method which requires a kind of HCI evaluation checklist, then that evaluation should have used a particular HCI evaluation checklist (i.e., an instance of the HCI evaluation checklist required by the adopted method. Thus, the following axiom applies:

A 6.3. $\forall$ *hcie*: HCI Evaluation, *m*: HCI Evaluation Method, *at*: HCI Evaluation Artifact Type follows(*hcie*, *m*) $\wedge$ requiresTheUseOf(*m*, *at*) $\rightarrow \exists$ *a*: HCI Evaluation Artifact uses(*hcie*, *a*) $\wedge$ isInstanceOf(*a*, *at*)

In the inspection-based case (Figure 6.1), Rita (**HCI Evaluator**) uses the Heuristic Evaluation (**HCI Evaluation Method**) to perform the evaluation (**HCI Evaluation**). Her goal in the evaluation (**HCI Evaluator Goal**) influenced her choice for that method (i.e., as her goal was to improve the usability (**HCI Quality Characteristic**) of the login UI (*User Interface*), she should use a method suitable for that purpose). Rita followed the Heuristic Evaluation method, which determines the 10 usability heuristics (**HCI Evaluation Criteria**) to be applied in the evaluation and requires the use of a checklist (**HCI Evaluation Artifact Type**). During the evaluation, Rita used a particular checklist (**HCI Evaluation Artifact**) that specifies each one of the 10 Nielsen's usability heuristics.

In some **HCI Evaluations** it is necessary to evaluate the *Interactive Computer System* during the *Human-Computer Interaction*. In this case, the *User* (or user representatives) of the *Interactive Computer System* participates in the **HCI Evaluation** while, at the same time, participates (*User Participation*), together with the *Interactive Computer System* (*Interactive Computer System Participation*), in the *Human-Computer Interaction*. When the *User* participates in an **HCI Evaluation**, it is required that this *User* uses the *Interactive Computer System* in order to achieve a certain goal (*User Goal*) (e.g., send an email). While the *User* interacts with the *Interactive Computer System* to achieve his/her goal (*User Goal*), the **HCI Evaluator** performs the **HCI Evaluation**, following the **HCI Evaluation Method** and applying **HCI Evaluation Criteria** to evaluate **HCI Quality Characteristics** of the *Interactive Computer System*. This scenario involves some constraints, addressed by the following axioms:

A 6.4. $\forall$ *hcie*: HCI Evaluation, *ics*: Interactive Computer System, *icsp*: Interactive Computer System Participation, *hci*: Human-Computer Interaction, *user*: User evaluates(*hcie*, *ics*) $\wedge$ participatesIn(*user*, *hcie*) $\wedge$ composedOf(*hci*, *icsp*) $\rightarrow \exists$ *up*: User Participation participationOf (*user*, *up*) $\wedge$ composedOf(*hci*, *up*)

A 6.5. $\forall$ *hcie*: HCI Evaluation, *hci*: Human-Computer Interaction, *hciqc*: HCI Quality Characteristic, *ics*: Interactive Computer System, *icsp*: Interactive Computer System Participation evaluates(*hcie*, *hci*) $\wedge$ evaluates(*hcie*, *hciqc*) $\wedge$ has(*ics*, *hciqc*) $\wedge$ composedOf(*hci*, *icsp*) $\wedge$ participationOf(*icsp*, *ics*) $\rightarrow$ evaluates(*hcie*, *ics*)

Moreover, in an **HCI Evaluation** an **HCI Evaluator** can act as a *User*. Conversely, a *User* can act as an **HCI Evaluator**. Thus, if an evaluator performs an evaluation in which he/she acts as a user, then this evaluator is also the user who participates in the evaluation. On the other hand, if a user that participates in an evaluation performs that evaluation, then he/she is also an evaluator of that evaluation. Hence, the following axioms apply:

A 6.6. $\forall$ *e*: HCI Evaluator, *hcie*: HCI Evaluation performs(*e*, *hcie*) $\wedge$ participatesIn(*e*, *hcie*) $\rightarrow \exists$ *u*: User participatesIn(*u*, *hcie*) $\wedge$ (*e* = *u*)

A 6.7. $\forall$ *u*: User, *hcie*: HCI Evaluation participatesIn(*u*, *hcie*) $\wedge$ performs(*u*, *hcie*) $\rightarrow \exists$ *e*: HCI Evaluator performs(*e*, *hcie*) $\wedge$ (*e* = *u*)

After the **HCI Evaluation** is performed, the **HCI Evaluation Report**, a Document, is created, under the responsibility of the **HCI Evaluator**, aiming to report the evaluation results and other relevant information, such as the applied **HCI Evaluation Criteria**. The **HCI Evaluation Report** is a Document, i.e., any written or pictorial information usually presented in a predefined format. Therefore, it can refer to, for example a description of the results registered in a textual form or a slides presentation, among others. If an **HCI Evaluation** has more than one **HCI Evaluator**, at least one of them must be responsible for the **HCI Evaluation Report**. Thus:

A 6.8. $\forall$ *hcie*: HCI Evaluation, *er*: HCI Evaluation Report creates(*hcie*, *er*) $\rightarrow \exists$ *e*: HCI Evaluator performs(*e*, *hcie*) $\wedge$ responsibleFor(*e*, *er*)

There are **HCI Evaluations** in which it is necessary to perform measurements to determine the extent to which **HCI Quality characteristics** of an *Interactive Computer System* meet **HCI Evaluation Criteria**. This is addressed in the conceptual model shown in Figure 6.5, which depicts a view of HCIEO that reuses concepts from COM to represent evaluations that quantify **HCI Quality Characteristics**. As in Figure 6.4, at the top we have UFO. At the center, there are COM (SEON Core Layer), containing core concepts related to measurement, and HCIO (HCI-ON Core Layer), providing HCI core concepts. At the bottom, there are HCIEO concepts, representing a complementary view of Figure 6.4 to complement the HCIEO conceptual model.

In the HCIEO and HCIO conceptual models previously shown, Interactive Computer System and User Interface are Objects in UFO while Human-Computer Interaction and User Participation are Actions. In COM (Barcellos; Falbo; Frauches, 2014), both Object and Action are subtype of Measurable Entity. Thus, in Figure 6.5 the generalization relations from Measurable

Entity shown in light gray are just to illustrate that being Objects or Actions in UFO, Interactive Computer System and User Interface, Human-Computer Interaction and User Participation also are Measurable Entities. Moreover, in the model shown in Figure 6.5, we focus on HCI Quality Characteristics that are quantified by some Measure. Thus, the generalization relation from Measurable Element shown in light gray is just to illustrate that in that context, HCI Quality Characteristic is a subtype of Measurable Element.

Besides being *Objects* (grounding in UFO), *Interactive Computer System* and *User Interface* are also Measurable Entities, i.e., entities that can be measured. In this sense, **HCI Quality Characteristic** is subtype of Measurable Element, i.e., a measurable property that characterizes *Interactive Computer System* (or a *User Interface*). Measures can be used to quantify **HCI Quality Characteristics**. For example, usability (**HCI Quality Characteristic**) characterizes a system (*Interactive Computer System*) or its UI (*User Interface*) and can be quantified, among others, by means of the number of wrong clicks or touches (Measure). **HCI Evaluation Criteria** can refer to Measures used to quantify **HCI Quality Characteristic**. For example, the criteria "the number of wrong clicks or touches must be smaller than two", used to evaluate usability, refers to the measure number of wrong clicks or touches, which, in turn, is used to quantify usability. In this context, the Measure referred by an **HCI Evaluation Criteria** in an **HCI Evaluation** that evaluates an **HCI Quality Characteristic** must be a Measure used to quantify that **HCI Quality Characteristic**. Thus:

A 6.9. $\forall$ *m*: Measure, *hciec*: HCI Evaluation criteria, *hcie*: HCI Evaluation, *hciqc*: HCI Quality Characteristic evaluates(*hcie*, *hciqc*) $\wedge$ applies(*hcie*, *hciec*) $\wedge$ usedToEvaluate(*hciec*, *hciqc*) $\wedge$ refersTo(*hciec*, *m*) $\rightarrow$ quantifies(*m*, *hciqc*)

A Measure (e.g., time spent to log in the system) can be expressed in a Measure Unit (e.g., minute) and has a Scale partitioned according to the Measure Unit and composed of the values that can be associated to the Measure. Measurement consists in collecting Measured Values to a Measure (e.g., the measurement of the time to log in the system, resulting in the value 1 minute). In an **HCI Evaluation**, Measurements are performed to establish Measured Values to quantify **HCI Quality Characteristics**.

In the user-based case (Figure 6.2), as in the inspection-based case, the **HCI Evaluation** evaluated the usability (**HCI Quality Characteristic**) of the web site login UI (*User Interface*). However, in the user-based case, Rita used measures to quantify usability (e.g., number of calls for help, number of wrong clicks or touches) and performed measurements to collect values to the measures (e.g., Alex gave two wrong clicks and called for help three times). After that, Rita considered the **HCI Evaluation Criteria** related to usability and referring to the used measures (e.g., "the number of wrong clicks or touches must be smaller than two" is the criteria related to the measure number of wrong clicks or touches) and the measured values to evaluate usability. When a **HCI Evaluation** involves measurements, the **HCI Evaluation Report** must

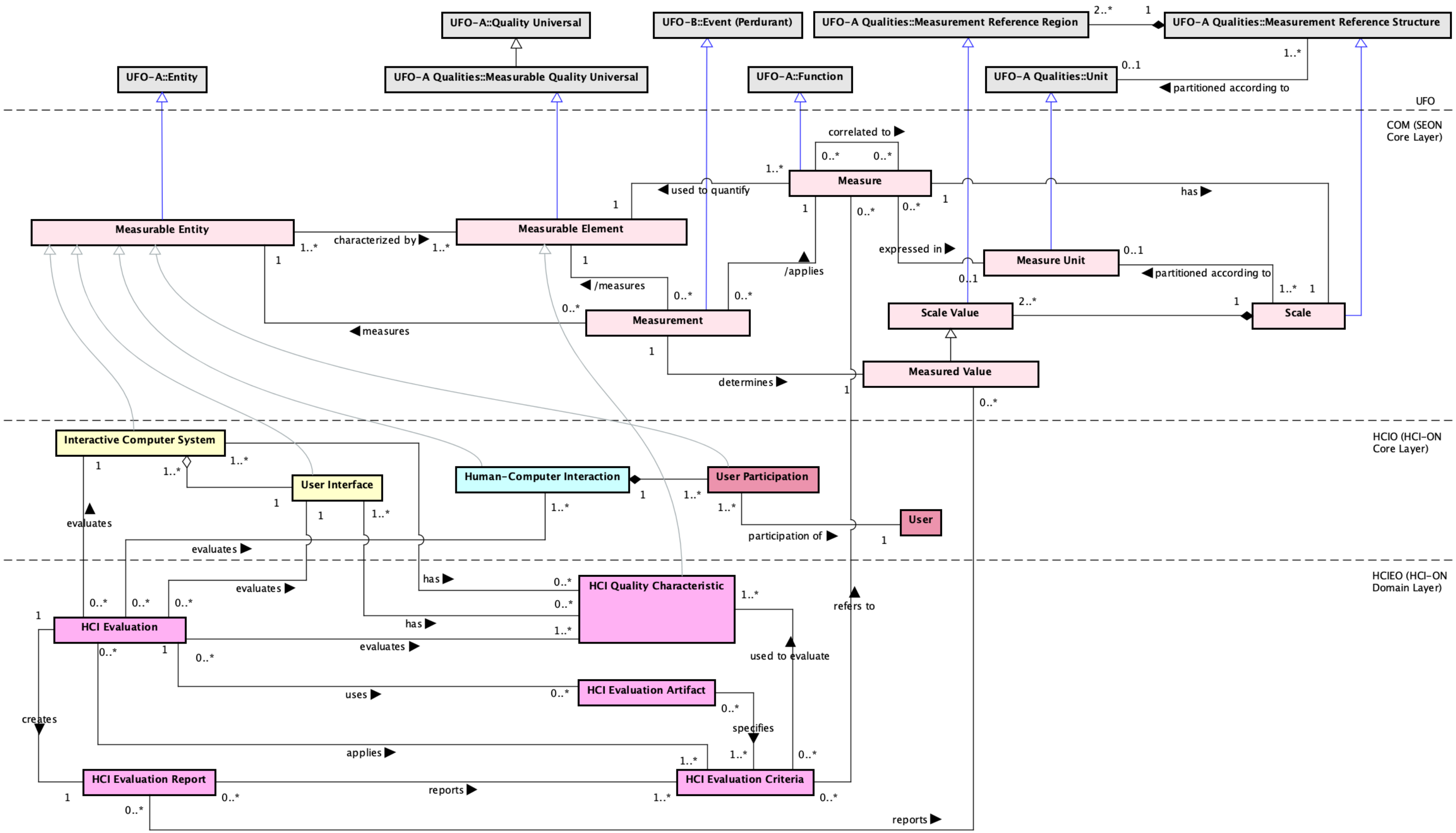


Figure 6.5 – HCIEO complementary view.

consider the Measured Values.

6.3 HCIEO Evaluation

Analogous to HCIO evaluation (Section 5.3), HCIEO was evaluated in Verification & Validation activities, as suggested in SABiO (Falbo, 2014). Also, the ontology specification was peer-reviewed by one domain expert. Based on the evaluation results, we improved the ontology and reached the current version. Some situations the domain expert pointed out as not properly covered by the previous version of HCIEO include: in an evaluation, there may be an observer, who does not necessarily have to be the person performing the evaluation; the observer can be a person or a system; the same person can play the role of evaluator and observer in an evaluation; in evaluations, with several evaluators, there must be at least one evaluator responsible for the evaluation report; the artifact used in an evaluation depends on the evaluation method adopted, and therefore, the method requires the use of a specific type of artifact. Feedback like this helped us to improve the ontology. Next, we present some results of V&V activities related to HCIEO.

6.3.1 Verification by Experts

During Verification, we manually answered the competency questions (CQs) and checked if HCIEO concepts are the ones sufficient and necessary to answer the CQs. Table 6.2 illustrates the results of the verification activity, showing which elements of HCIEO (concepts, relations and properties) answer the CQs. Verification results showed that HCIEO is able to answer the competency questions (i.e., the ontology addresses the established scope) and that it contains the sufficient and necessary elements to do so.

Table 6.2 – HCIEO verification against its competency questions.

CQ Id	Description, Concepts and *Relations*	**Axioms**
CQ01	**What leads to an HCI evaluation?** HCI Evaluator Goal is the *propositional content of* HCI Evaluator Intention, which *inheres in* HCI Evaluator and *causes* HCI Evaluation	
CQ02	**What does an HCI evaluate?** HCI Evaluation *evaluates* Interactive Computer System (or the User Interface that *composes* Interactive Computer System) and Human-Computer Interaction	A6.5
CQ03	**What quality characteristics are evaluated in an HCI evaluation?** HCI Evaluation *evaluates* HCI Quality Characteristic of the Interactive Computer System (or the User Interface that *composes* Interactive Computer System) *evaluated* in the HCI Evaluation	A6.1, A6.2
CQ04	**What criteria are applied in an HCI evaluation?** HCI Evaluation *applies* HCI Evaluation Criteria which is *used to evaluate* HCI Quality Characteristic of the Interactive Computer System (or the User Interface that *composes* Interactive Computer System) *evaluated* in the HCI Evaluation HCI Evaluation Criteria *refers to* User Requirements and Measure	
CQ05	**What artifacts are used to perform an HCI evaluation?** HCI Evaluation *uses* HCI Evaluation Artifact that *specifies* HCI Evaluation Criteria HCI Evaluation Artifact is *instance of* HCI Evaluation Artifact Type HCI Evaluation Method *requires the use of* HCI Evaluation Artifact Type	A6.3

Table 6.2 – *Continued from previous page*

CQ Id	Description, Concepts and *Relations*	Axioms
CQ06	**What is the result of an HCI evaluation?** HCI Evaluation *creates* HCI Evaluation Report	
CQ07	**What does the result of an HCI evaluation reports?** HCI Evaluation Report *reports* HCI Evaluation Criteria and Measured Values *determined* in Measurements that *measures* HCI Quality Characteristic	
CQ08	**Who performs an HCI evaluation?** HCI Evaluator *performs* HCI Evaluation and is *responsible for* HCI Evaluation Report	A6.6, A6.7, A6.8
CQ09	**Who participates in an HCI evaluation?** User *participates in* an HCI Evaluation HCI Observer *participates in* an HCI Evaluation	A6.4
CQ10	**What method is used in an HCI evaluation?** HCI Evaluation *follows* HCI Evaluation Method. HCI Evaluation Method is *chosen according to* the HCI Evaluator Goal and *can determine* HCI Evaluation Criteria *applied* in an HCI Evaluation	
CQ11	**How can quality characteristics be quantified?** HCI Quality Characteristic is *subtype of* Measurable Element which is *quantified* by Measure that is *expressed in* Measure Unit and *has* Scale *partitioned according to* Measure Unit	A6.9
CQ12	**What are the values assigned to quality characteristics of an interactive computer system in an HCI evaluation?** Interactive Computer System (or the User Interface that *composes* Interactive Computer System) is *subtype of* Measurable Entity and *has* HCI Quality Characteristic which is *subtype of* Measurable Element and is *quantified by* Measure Measurement *measures* HCI Quality Characteristic and *determines* Measured Value	

6.3.2 Validation

Ontology validation regards the capability of the ontology to properly represent real-world situations (Falbo, 2014). Hence, we instantiated the ontology using data extracted from the inspection-based and user-based cases. In Section 6.2 we used some instances from these cases when describing HCIEO conceptual model. Next, in Table 6.3 and in Table 6.4, we present the complete instantiation using, respectively, the cases illustrated in Figure 6.1 and Figure 6.2.

Table 6.4 – HCIEO Instantiation - user-based evaluation case.

Concept	Instance
HCIEO	
HCI Evaluation	Rita's action to evaluate the web site login user interface by observing and measuring the user interacting with the system
HCI Evaluation Artifact Type	Table
HCI Evaluation Artifact	A table to record values measured during the evaluation
HCI Evaluation Criteria	EC1 (referring to M1) - number of requested helping should be zero EC2 (referring to M2) - number of wrong clicks or touches should be up to 2 EC3 (referring to M3) - Time spent by the user on the first attempt to retrieve the password should be up to two minutes EC4 (referring to M4) - Time spent by the user on the second attempt to recover the password should be up to one minute
HCI Evaluator Goal	Complement the previous evaluation (Heuristic Evaluation) aiming at a students' web site login UI more usable and understandable
HCI Evaluation Method	Usability Testing
HCI Evaluation Report	Report created after the evaluation, containing the evaluation results considering measured values evaluated according to the considered evaluated criteria

Table 6.4 – *Continued from previous page*

Concept	Instance
HCI Evaluator/HCI Observer (Agent HCI Observer)	Rita
HCI Evaluator Intention	Find usability problems in the web site login UI
HCI Quality Characteristic / Measurable Element	Usability
HCIO	
Interactive Computer System / Measurable Entity	The web system used by the students
User Interface / Measurable Entity	The login UI of the web system used by students
User	Alex
User Goal	Recover his password
COM	
Measure	M1 - number of calls for help M2 - number of wrong clicks or touches M3 - time to recover the password on the first use M4 - time to recover the password on the second use
Measured Value	MV1 (related to M1) - three calls for help MV2 (related to M2) - two error clicks MV3 (related to M3) - Three minutes to recover the password on first use MV4 (related to M4) - One and a half minutes to recover the password on the second use

The HCIEO dictionary of terms is available at (Costa; Barcellos, 2021). Moreover, HCIEO's operational version is under development.

6.4 Discussion

In this section we make some discussions about HCIEO conceptualization.

HCIEO conceptualization aims to define what an evaluation is within the HCI scope, following the approach of human-centered design. Thus, HCIEO addresses, in the user-centered evaluation perspective, HCI evaluation, produced and used artifacts, considered criteria, evaluated quality characteristics, adopted methods, user participation and qualitative and quantitative evaluation.

To simplify the conceptual model of HCIEO, we decided to adopt the term 'evaluates' for the evaluative nature of relations arising from an HCI Evaluation. According to the need of the evaluation context (method used) in which the HCI Evaluation refers, this term can be replaced by terms such as 'inspects'. For example, in inspection-based evaluations, the term 'evaluates' should be read as 'inspects'.

As showed by instantiating the cases presented in section 6.1, the ontology allows representing different ways of evaluating the quality characteristics of an interactive computer system, covering cases where the evaluator performs the evaluation without user participation (Figure 6.1) as well as cases where there is the user's participation during the evaluation (Figure

Table 6.3 – HCIEO Instantiation - inspection-based evaluation case.

Concept	**Instance**
HCIEO	
HCI Evaluation	Rita's action to inspect the web site login user interface
HCI Evaluation Artifact Type	Checklist
HCI Evaluation Artifact	A checklist containing Nielsen's usability heuristics
HCI Evaluation Criteria	Each one of the 10 Nielsen's usability heuristics (e.g., #1: Visibility of system status; #2: Match between system and the real world; #3: User control and freedom; #4: Consistency and standards; #5: Error prevention; #6: Recognition rather than recall; #7: Flexibility and efficiency of use; #8: Aesthetic and minimalist design; #9: Help users recognize; diagnose; and recover from errors; and #10: Help and documentation.)
HCI Evaluator Goal	Make the students' web site login UI more usable and understandable
HCI Evaluation Method	Heuristic Evaluation
HCI Evaluation Report	Report created after the evaluation, containing the evaluation results according to the considered evaluated criteria
HCI Evaluator/HCI Observer (Agent HCI Observer)	Rita
HCI Evaluator Intention	Find usability problems in the web site login UI
HCI Quality Characteristic	Usability
HCIO	
Interactive Computer System	The web system used by the students
User Interface	The login UI of the web system used by the students

6.2). HCIEO also points out the need of using evaluation methods. They are crucial procedures to perform an evaluation and, therefore, must be followed by the evaluator during an evaluation.

Although there are several factors (such as available time and resources, possibility of recruiting users or user representatives, cost benefit and so on) that need to be considered to select a method suitable for an HCI evaluation, it is important to take the evaluator goal into account as the first premise to be considered. HCIEO also allows more than one evaluation method to be adopted in an HCI evaluation. When possible, the use of multiple methods is recommended in the literature because it allows a more complete view of the system's quality characteristics, since each method evaluates them in different ways. Multiple methods can be adopted in the same evaluation or several evaluations adopting different methods can be carried out to evaluate the same interactive computer system.

According to the literature and standards, certain evaluations can be performed by the user of the system to be evaluated. Therefore, there are evaluation methods that the user can follow to evaluate HCI quality characteristics of an interactive computer system. This circumstance is treated in HCIEO by means of an axiom, which states that if a user performs an evaluation in which he/she step through playing the role of an evaluator, then this user is also the evaluator who performs the evaluation. On the other hand, when there are no resources available to recruit users to participate in the evaluation, it is possible to adopt evaluation methods (e.g., usability-walkthrough) that allow the evaluator to step through a scenario (interaction) playing the role of user and, at the same time, act as an evaluator by identifying problems associated with the successful completion of the scenario. This is also treated by HCIEO through the axiom that says that if an evaluator performs an evaluation in which he/she step through playing the role of a user, then this evaluator is also the user who participates in the evaluation.

HCIEO encompasses qualitative (as presented in the inspection-based case, Figure 6.1) and quantitative evaluation (as presented in the user-based case, Figure 6.2). Quantifying the quality characteristics to evaluate interactive computer system is possible due to the anchoring of HCIEO in concepts of the Core Ontology on Measurement (COM).

HCIEO also deals with evaluation criteria, which can be determined by evaluation methods or defined according to the evaluation needs. In the heuristic evaluation scenario (qualitative evaluation), heuristics determined by the method were used by Rita to evaluate the system, while in the usability testing (quantitative evaluation) the criteria were defined in a quantitative way, according to the evaluator goal.

As discussed in Section 2.1.2.2, in an evaluation, data collection can be performed by a person (direct observation) or by a system (indirect observation). HCIEO covers these scenarios by considering two different roles: the evaluator (HCI Evaluator) and the observer (HCI Observer). The observer, which can be an agent (Agent HCI Observer) or a system (Computer HCI Observer), gets data through observation (inspection, etc.) for an HCI Evaluation In the cases exemplified in this chapter, we presented direct observation, where Rita plays the roles of evaluator and also observer (when getting data). In Section 7, a real case of indirect observation, whose observer is a system, is presented and discussed.

Finally, it is important to say that HCIEO does not aim to deal with the evaluation process (e.g., the activities that compose an HCI evaluation and the order in which they are performed). These aspects should be addressed by a specific domain ontology devoted to the HCI evaluation process, which should reuse concepts from HCIEO and be integrated to HCI-ON.

6.5 Comparing HCIEO to other Ontologies

As we mentioned in the Introduction of this chapter, in our secondary study (Chapter 3) we identified five ontologies (#16, #20, #22, #28 and #32) partially related to HCI evaluation. When analyzing these ontologies, we noticed that they were not developed to completely cover the HCI evaluation domain. Most of them were developed to meet specific purposes (# 16, Personas; # 20, UI ergonomics; # 22, User's feedback; # 28, Pervasive Game) and, therefore, the evaluation part is dealt with in a few concepts present in the ontologies. Although # 32 also deals with a specific subject, Web UI Measurement, it is closer to HCI evaluation.

Briefly, #16 addresses user-centered design methods, namely Personas and Usability Test. However, its conceptual model emphasizes Personas, even when dealing with Usability Test. Hence, in this ontology, Usability Test is exclusively related to Personas and is covered by only one concept (UsabilityTest). #20 presents the field of ergonomics of HCI. The focus of the ontology is on user profiles and customized interfaces according to ergonomic recommendations. In its conceptual model there are only three concepts related to HCI evaluation (Design_And_-Evaluation, User_Evaluation, Interface_Evaluation). The HCI evaluation part is focused on the questionnaire method, but the method is not addressed explicitly in the ontology. #22 focuses on the description and structure of user feedback. In this structure, there is the Evaluating concept, which aims to represent user feedback as its evaluation of the system using a goodness/badness scale. Thus, in this ontology, the HCI evaluation aims to represent the questionnaire method, which allows the user to present his/her point of view regarding the system and its performance after the interaction phenomenon. #28 covers aspects related to pervasive games (PG) from the player experience perspective. It was developed with the aim of providing a consensus vocabulary or instrument to talk about pervasive games from the user experience perspective. In its conceptual model there is a relationship between Game and Metrics concepts. Metrics in this context are related to software metrics and was broken down into the following concepts: Motivation, Usability, Engagement, Playability, Effectiveness and Others. These concepts are the same in which pervasive experience is expressed. The part related to the metrics, user experience (UX) and HCI evaluation of this ontology is restricted to the context of game user experience evaluation. Lastly, #32 aims to structure web user interface (WUI) metrics. In this context, Metrics means UI's quantitative characteristics. In its conceptual model, the Metrics concept is directly related to usability and performance as well as interface and attribute. The ontology aims at the classification of WUI metrics.

In sum, none of the aforementioned ontologies were developed to address the HCI Evaluation domain as HCIEO. They approach this domain superficially through some concepts. We consider #16 the ontology most related to our proposal (HCIEO). Despite containing only one concept related to HCI evaluation (UsabilityTest), the conceptual model was built following the user-centered design, as well as in HCIEO. #16, #20 and #22 are the ones more related to HCI evaluation methods (#16 covers Usability Test method while #20 and #22 cover HCI evaluation questionnaire method). #28 and #32 are the ones we can relate to HCI evaluation metrics. The former (#28) covers metrics in the context of pervasive experience, while the later (#32) structures WUI metrics. HCIEO covers evaluation methods through a concept related to HCI Evaluation, not representing any specific method. Concerning metrics, due to the resuse of concepts from COM, HCIEO addresses the use of measures (that have the same meaning than metrics in #28 and #32) to quantify HCI quality characteristics.

Since HCIEO and the aforementioned ontologies have different coverages and purposes, it is difficult to make a fair comparison. Hence, the analysis we made considers quality characteristics that should be ensured during the ontology engineering process. Table 6.5 presents HCIEO and the five aforementioned ontologies considering the characteristics of "beautiful ontologies" (D'Aquin; Gangemi, 2011).

As it can be noticed, only HCIEO meets all the considered quality characteristics. Concerning *good domain coverage*, in the table we marked all ontologies in the sense that each of them covers the domain portion it is intended to. With respect to *modularity* only three (#16, #20 and #32) ontologies are modular (#16 uses schema; #20 uses part; and #32 uses top-level

Table 6.5 – Analysis of HCIEO and HCI-related ontologies.

"Beautiful ontology" criteria	HCI-related ontologies #16	#20	#22	#28	#32	HCIEO
good domain coverage	√	√	√	√	√	√
being modular	√	√			√	√
being formally rigorous						√
implementing also non-taxonomic relations	√			√	√	√
following an ontology engineering methods				√		√
following an evaluation method	√					√
reusing foundational ontologies						√
being based on competency questions				√		√
implementing an international standard					√	√
addressing HCI Evaluation	partially (method)	partially (method)	partially (method)	partially (metrics)	partially (metrics)	completely

concepts). As for *being formally rigorous*, none of them represent some degree of formalism, even if not very rigorous. Concerning *implementing also non-taxonomic relations*, only three (#16, #28, and #32) ontologies consider both taxonomic and non-taxonomic relations. As for *ontology engineering methods* only #28 ontology follows Methontology (Fernández-López; Gómez-Pérez; Juristo, 1997) and NeOn (Gómez-Pérez; Suárez-Figueroa, 2009). Regarding *following an evaluation method*, only #16 was evaluated thought a task-based evaluation. As for *reusing foundational ontologies*, none of them reuse foundational ontologies. With respect to *being based on competency questions*, only #28 uses competency questions do define its scope. Only #32 *implements an international standard*. Finally, with respect to *addressing HCI Evaluation*, three (#16, #20 and #22) ontologies address HCI evaluation method and the other two (#28 and #32) address HCI evaluation metrics. Thus, we consider that HCI-related ontologies partially address the HCI Evaluation domain. In conclusion, only HCIEO covers the main aspects of HCI evaluation and followed appropriate engineering method and practices that resulted in a consistent conceptualization.

6.6 Final Considerations of the Chapter

This chapter presented HCIEO, a domain reference ontology about HCI Evaluation. HCIEO scope was defined by means of competency questions as suggested by the adopted ontology engineering method. Concerning its architecture, HCIEO is grounded in UFO, reuses and extends concepts from SPO, SysSwO, COM, RSRO, HCIO and HCIDO.

Based on the secondary study results and in the analysis of the five ontologies related to

HCI evaluation (presented in Section 6.5), HCIEO is the first ontology that addresses evaluation in the HCI domain. HCIEO aims at a consensual and shared conceptualization in accordance with knowledge presented in the literature and in the existing standards. HCIEO contributes to the HCI area by providing a conceptualization of the HCI evaluation, which can be used for communication and learning purposes as well as to support knowledge-related and interoperability solutions, as we discuss in the next chapter.

7 HCI-ON Applications

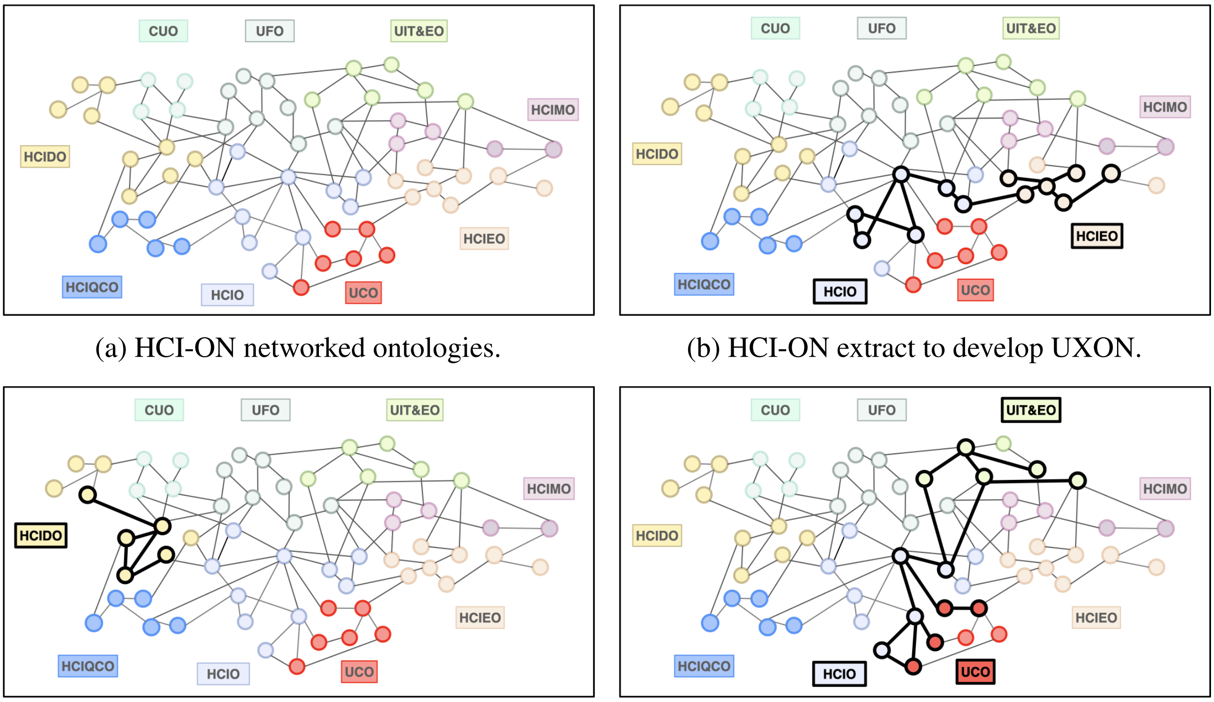
This chapter presents three practical applications of HCI-ON, which were performed in the *Design Cycle* to evaluate HCI-ON and demonstrate its use to solve HCI-related problems. The first application refers to the use of HCI-ON to develop a system that aids in the evaluation of user experience of an immersive application. In the second one, HCI-ON was used in the development of a knowledge-based system to help HCI design. In the third application, HCI-ON was used to build a social network with adaptive interface. The first application was developed in the context of this thesis and a undergraduate project (Manso, 2022). The other two were developed in partnership with other students, in the context of other master and doctorate research projects related to this work. The use of HCI-ON in these practical applications meets the requirement R9 (support interoperability and knowledge-related solutions in the HCI domain) and satisfies the evaluation criteria C1 (the knowledge framework must be able to represent real-world situations) and C3 (the knowledge framework must be useful to develop interoperability and knowledge-related solutions in HCI and its use must be viable). R9 and C3 are covered partially because the applications do not address interoperability problems. Section 7.1 introduces the chapter. Section 7.2 presents UXON (User eXperience Ontology Network-based system) and discusses how we evaluated it. Section 7.3 concerns KTID (Knowledge Supporting Tool For Human-Computer Interaction Design). Section 7.4 addresses SNOPI (Social Network with Ontology-based adaPtive Interface). Last, Section 7.5 closes the chapter.

7.1 Introduction

As we argued in the previous chapters, one benefit provided by ONs is the possibility of extracting portions of the represented knowledge according to the problem to be solved. In this sense, we used extracts of HCI-ON to develop three systems. By doing so, we demonstrate that HCI-ON can be used in a flexible way and that its use for supporting system development is feasible.

Figure 7.1 exemplifies flexibility in HCI-ON use. Suppose that Figure 7.1a represents all concepts and relations of the HCI-ON networked ontologies. Once the concepts are integrated in a consistent way, one can just select the fragment that reflects the domain to be treated. Figures 7.1b, 7.1c, and 7.1d illustrate different extracts used to develop the three systems addressed in this chapter.

In the next sections, we present the three systems developed using HCI-ON extracts, namely: User eXperience Ontology Network-based system (UXON), Knowledge Supporting Tool For Human-Computer Interaction Design (KTID) and Social Network with Ontology-based adaPtive Interface (SNOPI).



(a) HCI-ON networked ontologies.

(b) HCI-ON extract to develop UXON.

(c) HCI-ON extract to develop KTID.

(d) HCI-ON extract to develop SNOPI.

Figure 7.1 – HCI-ON extracts used in ontology-oriented software development.

7.2 User eXperience Ontology Network-based system - UXON

We have worked with the Usability and Software Engineering Research Group (USES Research Group[1]), which reported the need to evaluate the user behavior (particularly user experience) in his/her interaction with a mobile entertainment application for large events (Amazonas et al., 2019; Marques et al., 2020). The application, called *Compomus*, is an immersive technology used by many people and its goal is to create a sense of immersion for the user by transforming the audience's role from a mere spectator to an active element of the show (Amazonas et al., 2019; Marques et al., 2020). The user experience is measured by means of its engagement in the immersive interaction.

For evaluating the user behavior, it is necessary to collect data during the user interaction with the mobile application, use collected data to calculate user experience metrics and analyze them. Since interaction data regards many users and should be collected without interrupting the user experience, it is not feasible to collect and analyze data manually. Thus, an automated solution is needed.

The solution consists of using ontologies as a basis of a system that collects and stores data, as well as analyses data and presents consolidated information about user experience. The ontology is used both as conceptual model and operational ontology. The operational ontology, implemented in OWL[2], is used to capture interaction data recorded in logs. Data is stored in a

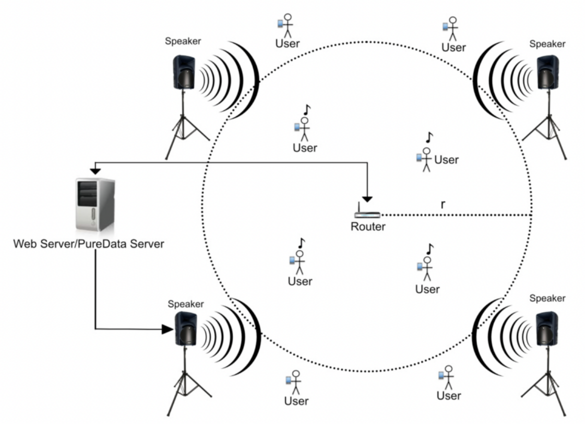
1 <http://uses.icomp.ufam.edu.br/index.php/en/>
2 <https://www.w3.org/OWL/>

triplestore built considering the ontology conceptual model. The operational ontology is then used to provide information about user experience through a set of metrics such as interactivity and user interaction (Marques et al., 2020).

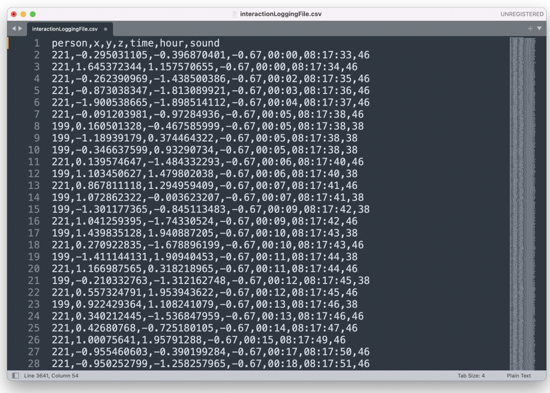
UXON was developed in the context of a undergraduate project (Manso, 2022) supervised by the author of this theses. UXON uses a fragment of HCI-ON that includes concepts from HCI Ontology (HCIO), HCI Evaluation Ontology (HCIEO) and also from the Core Ontology on Measurement (COM) of SEON (Barcellos; Falbo; Frauches, 2014).

7.2.1 Understanding the Problem

Compomus (Amazonas et al., 2019; Marques et al., 2020) is an immersive music composition application that collects interaction data (interaction logging) from various users during the collective production of music. The musical composition event is carried out in sessions that are configured in a time interval and in groups of people who occupy the same room, in which there are four speakers. Each person in the session uses Compomus on her cell phone to choose from 50 types of sound (Figure 7.2a). When a sound is chosen by one person, it is played on one of the four speakers, which simultaneously emit the sounds chosen by the other people participating in the session. The speakers are geographically positioned forming the musical environment (a rectangle) and people move within this environment, selecting/playing sounds on the speakers through interaction with Compomus (Figure 7.2a). For each person and each movement or sound choice in this environment, Compomus records in an interaction log file (Figure 7.2b) the following data: *person*, *x*, *y*, *z*, *time*, *hour* and *sound* (first line of Figure 7.2b).



(a) System Function Diagram (Amazonas et al., 2019).



(b) Compomus interaction log file.

Figure 7.2 – Compomus overview.

Person refers to the participant id. *X*, *y* and *z*[3] together refer to the *Person's geolocation* in the music composition environment. *Time* refers to the duration of the session until the data

[3] As the music composition environment is two-dimensional, and *z* refers to the third dimension in a dimension

record. *Hour* refers to the time the interaction took place. *Sound* refers to the sound chosen by the *Person*.

The UX metrics applied to understand the individual behavior of the user and evaluate Compomus UX evaluation were defined by Marques et al. (2020) and are described as follows:

- *User interactivity*: responsible for evaluating the interaction time of each participant, verifying the total time of an experience and relating it to the individual interaction time of users. The variables of this metric are: the overall session time (T_{sg}), the start (T_i) and end time (T_f) of the event, logoff time (T_{off}) and the user login time (T_{in}). This metric is calculated using the following formula:

$$T_{sg} = \frac{(T_f - T_i)}{(T_{off} - T_{in})}$$

- *User behavior*: responsible for evaluating the quality of the interaction time in terms of the engagement of each participant. In the case of Compomus, the sound change (*sound*) and the geolocation change (x, y) are considered. This metric is calculated using the following formula:

$$MC = \sum_{j}^{n} v$$

  The above formula is generic for the behavior metric. The sum indicates the number of interactions, v is the variable that represents the recorded interaction and the variation from j to n indicates the number of records.

- *Percentage of interactions*: responsible for investigating the behavior of users, using the participant with the highest number of interactions as a *benchmark* (100%) and analyzing the other participants in relation to this value. The variables of this metric are: the percentage of interactions of user u (P_u), the value of the metric of user behavior u and the value of the metric of behavior of the most active user b (*benchmark*) (P_b). This metric is calculated using the following formula:

$$P_u = \frac{MC_u}{MC_b}$$

7.2.2 UXON Overview

An overview of UXON is shown in Figure 7.3. Compomus captures data regarding the user interaction and records it in the interaction log file. The UX evaluator uploads the interaction

structure, despite being recorded in the log, it is not used as it does not reflect a interaction of geolocation change in the use of Compomus.

log file and then an ETL (Extract Transform and Load) process is performed using the HCI-ON extract to assign semantics to data. Data is stored in a triplestore and it is used to calculate metrics and provide other information, which are searched using SPARQL. The results can be visualized in different graphs and tables. The UX evaluator visualizes the results and analyzes them. Data and analysis results are recorded in an evaluation report. The ETL process, data persistence in the triplestore and SPARQL queries all make use of *ontoUXON*, the operational version of the HCI-ON extract used in the solution.

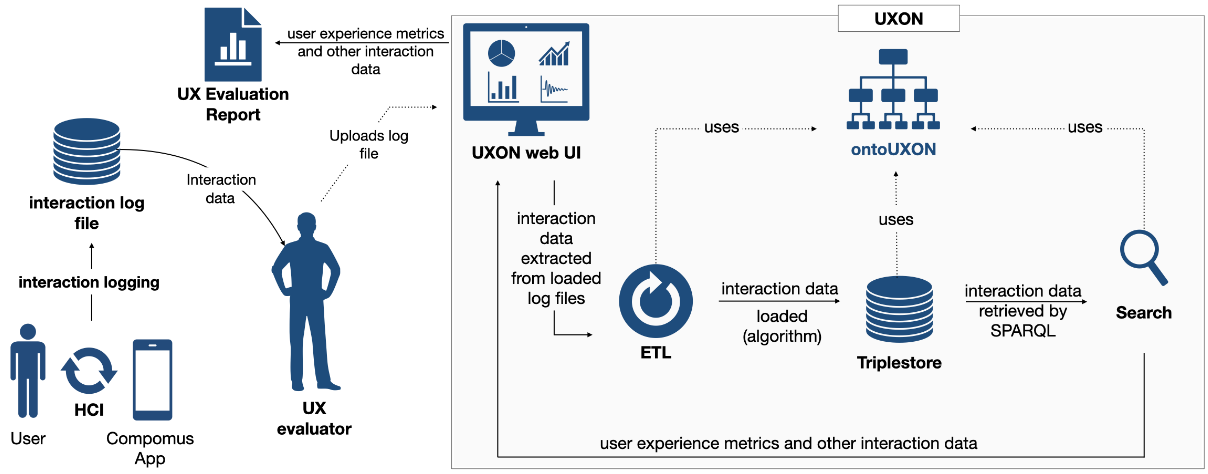


Figure 7.3 – UXON overview.

The development of UXON was based on the HCI-ON extract that will be discussed in the next section and followed the Ontology-Driven Development (ODD) and Ontology-Based Architectures (OBA) approaches (Happel; Seedorf, 2006). According to Happel and Seedorf (2006), ODD uses ontologies in development time to support and describe the problem domain itself, and OBA uses ontologies as primary artifacts in run-time, playing a major role in application logic.

The HCI-ON extract played a fundamental role. At development time (ODD), it contributed to understanding the application domain (i.e., UX evaluation) and defining UXON's business (translated into business rules and algorithms) and application logic. It also contributed to defining UXON's conceptual model and *ontoUXON* (presented in Section 7.2.3). *ontoUXON* is the HCI-ON fragment (reference conceptual model) transcribed into OWL (operational artifact).

At run-time (OBA), *ontoUXON* enabled the ETL process. As *ontoUXON* is an RDF graph/knowledge graph, it was used as the dataset (data model) in the UXON's triplestore configuration. Consequently, *ontoUXON* was used to express queries (SPARQL) across it. To put in another way, all data and also metrics values are instantiated in *ontoUXON* and later stored in the triplestore, which is searched by SPARQL queries.

In the next section we discuss the UXON's conceptual model and *ontoUXON*.

7.2.3 UXON's Conceptual Model and *ontoUXON*

To develop UXON, first we extracted the fragment of HCI-ON that addresses the problem domain. The HCI-ON extract considered to develop UXON is shown in Figure 7.4. It should capture the user actions and support to evaluate user experience. Thus, it includes aspects related to the HCI phenomenon (e.g., *User*, *User Participation*, *Human-Computer Interaction*), HCI evaluation (e.g., *HCI Evaluation*, *HCI Evaluation Report*, *HCI Evaluator*) and measurement (e.g., *Measure*, *Measurement*, *Measured Value*, among others).

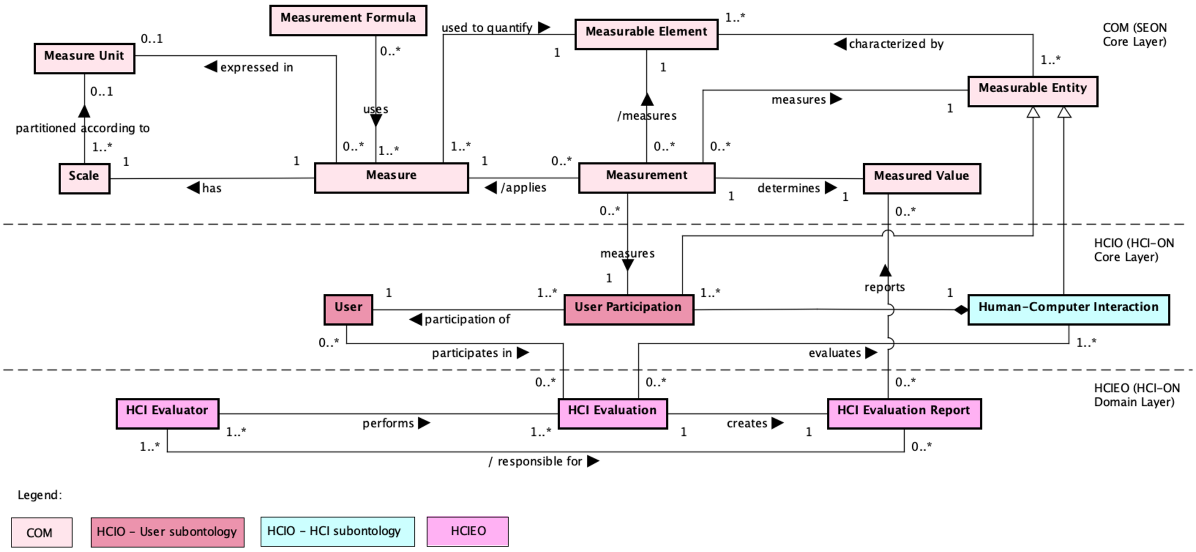


Figure 7.4 – HCI-ON's extract use to develop UXON.

Then, we made some adjustments in the conceptual model to turn it more suitable for implementation. In summary: (i) we do not represent the Measurable Entity concept because the only entity measured in UXON is User Participation (with that, the *measures* relation between Measurable Entity and Measurement is represented between Measurement and User Participation); (ii) the relationship *is quantified by* between User Participation and Measure was created (even though this information can be obtained from the relationships between User Participation, Measurement and Measure); (iii) Scale Value, Measure Unit, Measurement Formula, and Measurable Element[4] concepts are represented as attributes of Measure; (iv) the Measured Value concept is represented as a Measurement attribute; (v) attributes were created to store data such as the evaluator's name and its comments resulting from data analysis. We also adjusted the model to make it able to store data specific to the problem domain. For that, we defined new attributes to User Participation and Human-Computer Interaction to store data about the user interaction when using Compomus (e.g., user participation geolocation, sound and the interaction time). Details about the changes made in the conceptual model are presented in (Manso, 2022). Figure 7.5 shows the resulting conceptual model.

[4] For simplification reasons, although the same measurable element can be quantified by more than one measure, in UXON, a measurable element (e.g., behavior, interactivity) is treated in only one measure.

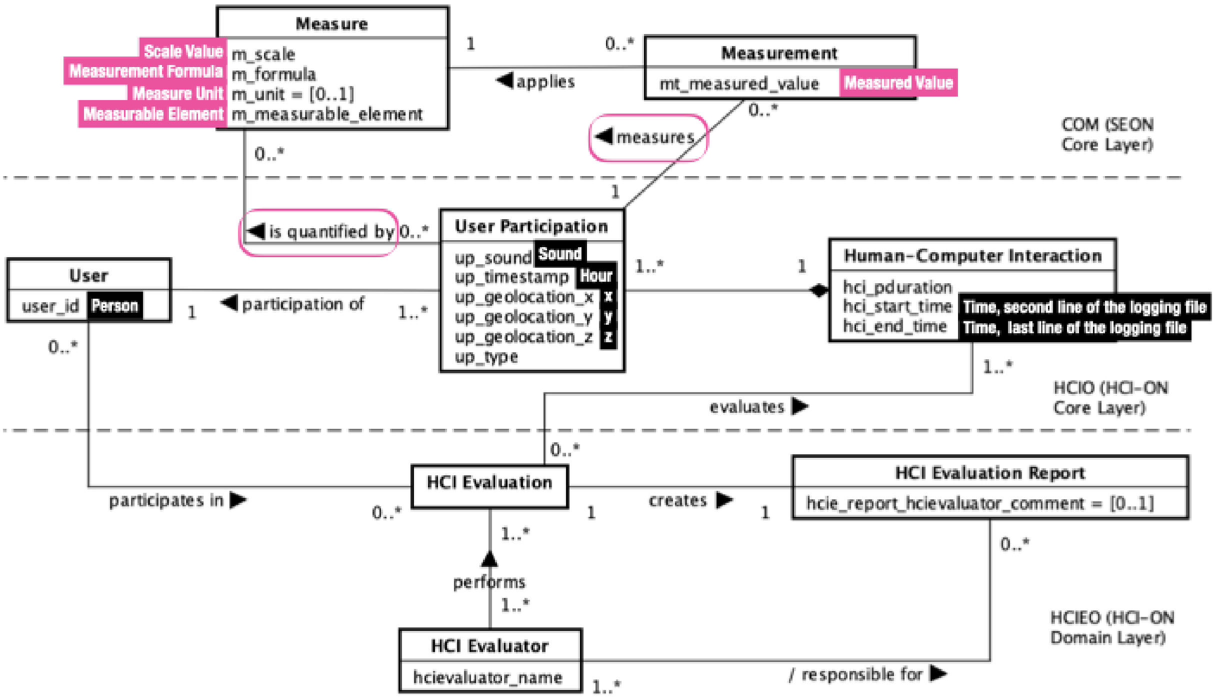


Figure 7.5 – UXON's conceptual model.

From UXON's conceptual model, we created *ontoUXON*, by transcribing the model to OWL using the Protegé[5]. Figures 7.6a, 7.6b and 7.6c present a fragment of *ontoUXON*. Semantic Web technologies (OWL, RDF, etc) allow representing knowledge in RDF triple [Subject → Predicate (or "Property") → Object] and RDF graph. Figure 7.6d represents these fragments in the form of a RDF graph (also known as knowledge graph). RDF graphs are used as database (triplestore, also known as subject-predicate-object databases). *ontoUXON* is available at <https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#>.

Technical information about UXON development is available at (Manso, 2022). UXON's source code is available at <https://github.com/cfmanso/uxon-final>.

7.2.4 UXON Features

In this section, we present some of the UXON features by showing some screenshots. UXON is available at <https://dev.nemo.inf.ufes.br/uxon/>.

Figure 7.7 shows the page where the UX evaluator uploads the log file of a session of Compomus.

From data loaded from the log, UXON calculates values for metrics and presents them in tables and graphs. Figure 7.8 illustrates a table and a graph showing data regarding *interactivity*, *behavior* and *percentage of interactions*. Figure 7.9 shows the "Top 5" graphs, which depict the 5 most emitted sounds and the 5 most active users in the session.

[5] Protégé is a free and open source ontology editor. The version used was 5.5.0.

```
<!--  https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#Measurement  -->
<owl:Class rdf:about="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#Measurement"/>
<!--  https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#User  -->
<owl:Class rdf:about="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#User"/>
<!--  https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#UserParticipation  -->
<owl:Class rdf:about="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#UserParticipation"/>
```

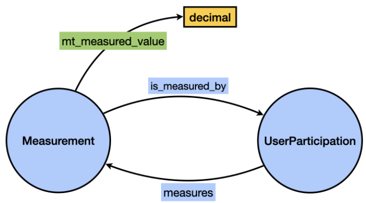
(a) Measurement, User and UserParticipation classes.

```
<!--  https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#is_measured_by  -->
▼<owl:ObjectProperty rdf:about="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#is_measured_by">
  <owl:inverseOf rdf:resource="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#measures"/>
  <rdfs:domain rdf:resource="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#UserParticipation"/>
  <rdfs:range rdf:resource="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#Measurement"/>
</owl:ObjectProperty>
```

(b) Object Property: is_measured_by.

```
<!--  https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#mt_measured_value  -->
▼<owl:DatatypeProperty rdf:about="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#mt_measured_value">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="https://dev.nemo.inf.ufes.br/hcion/ontoUXON.owl#Measurement"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
</owl:DatatypeProperty>
```

(c) Data Property: mt_measured_value.



(d) RDF graph.

Figure 7.6 – *ontoUXON*'s fragment.

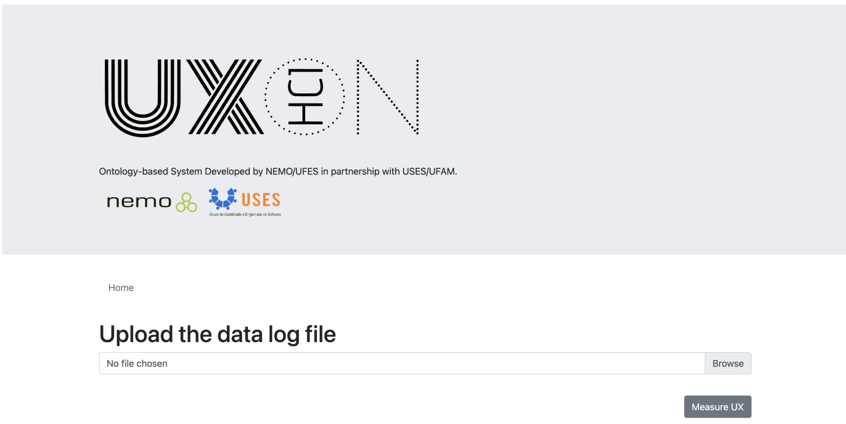


Figure 7.7 – UXON's home page.

UXON also shows geolocation maps (Figure 7.10), which provide information about the movements made by the users during the session. The maps can be viewed from a static or dynamic perspective.

The UX evaluator can also perform additional searches. There are some defined queries (e.g., the evaluator can search for user interactions that involved sound change or for how many times a sound was chosen by the users) and he/she can also create new queries. Figure 7.11 shows the page where the evaluator accesses defined queries and Figure 7.12 depicts the page

Figure 7.8 – Screenshots showing values calculated for UX metrics.

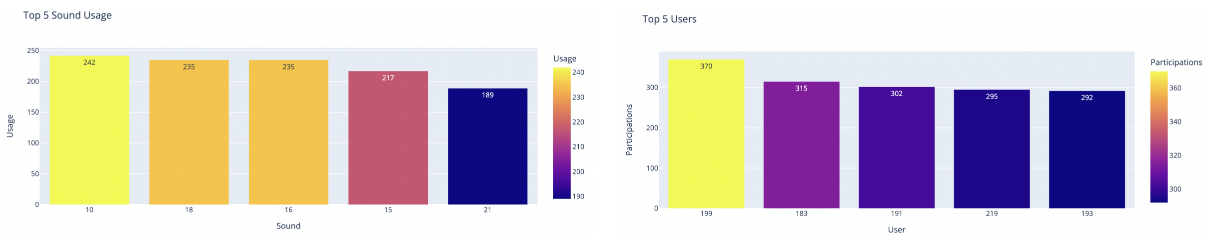


Figure 7.9 – Graphs showing Top 5 sounds and users.

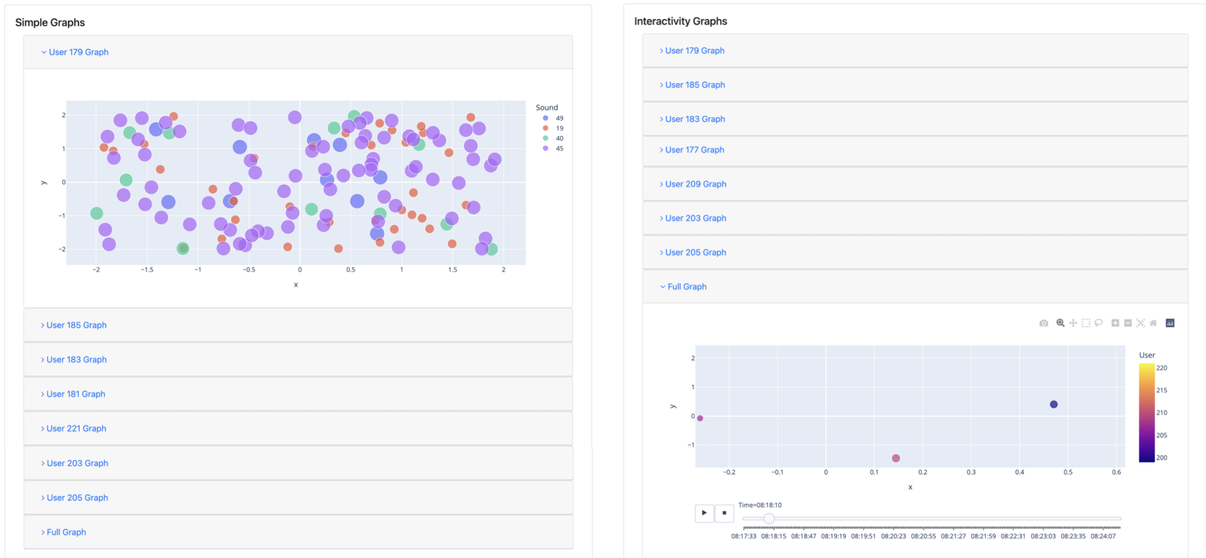


Figure 7.10 – Geolocation graphs showing how one or all users moved during the session.

where he/she can create new queries.

Finally, after analyzing data about the user interactions with Compomus, the UX evaluator records his/her conclusions about Compomus UX in an Evaluation Report, as shown in Figure 7.13. The evaluation report containing all tables, graphs and the evaluator comments can be downloaded.

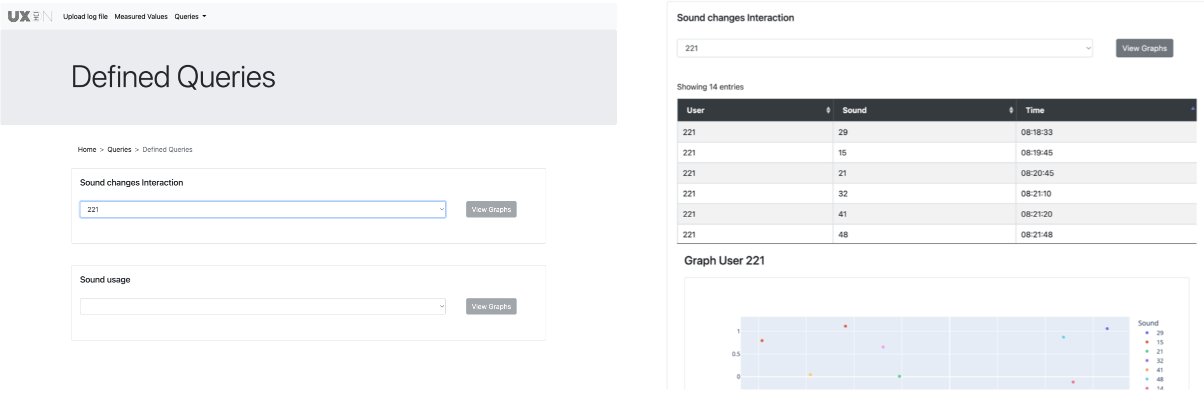


Figure 7.11 – Defined Queries page.

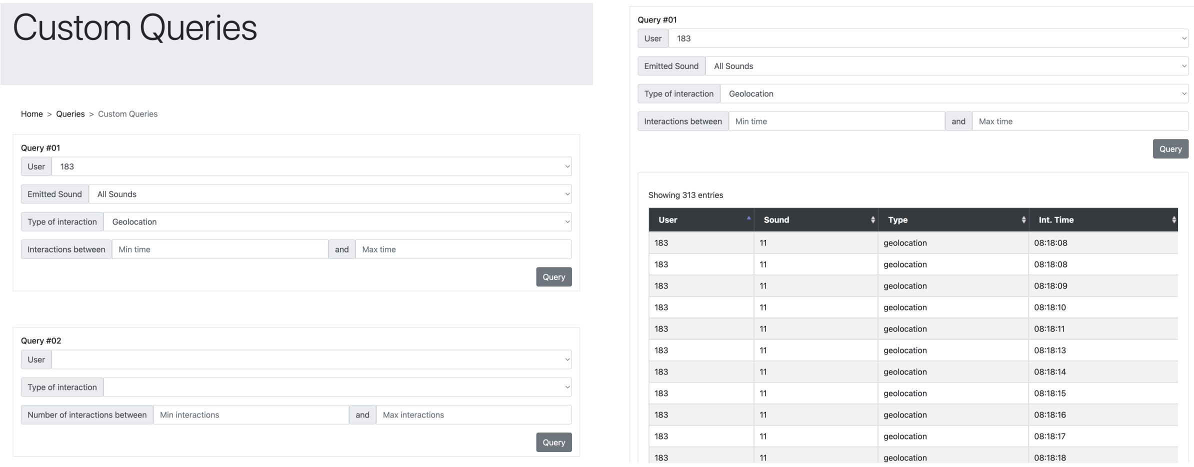


Figure 7.12 – Custom Queries page.

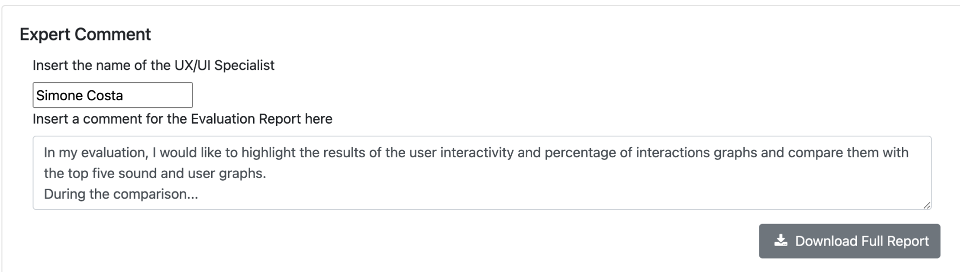


Figure 7.13 – Evaluation Report generation.

7.2.5 UXON Evaluation

The use of an HCI-ON extract to develop UXON served as a proof of concept to demonstrate that it is feasible to apply HCI-ON to develop systems for solving HCI problems. Aiming to evaluate if the solution produced by using HCI-ON is suitable for solving the aimed HCI problem, we performed a study and applied a questionnaire to three UXON users to get their perception about UXON. In addition, to obtain feedback about using HCI-ON to develop UXON, we performed an interview with the UXON developer. These studies allowed us to evaluate the use of HCI-ON from two perspectives: user (Section 7.2.5.1) and developer (Section 7.2.5.2).

7.2.5.1 UXON Evaluation – User Perspective

This section presents the study carried out with UXON users, which enabled us to find preliminary evidence to evaluate and improve UXON in terms of its usefulness and the feasibility of using it. Section 7.2.5.1.1 presents the study planning; Section 7.2.5.1.2 addresses its execution and main results; in Section 7.2.5.1.3, we discuss the results; and Section 7.2.5.1.4 addresses the threats to validity.

7.2.5.1.1 Study Planning

The study ***goal*** was to verify whether (an extract of) HCI-ON can be used to produce a suitable solution for an HCI-related problem. Following the GQM approach (Basili; Caldiera; Rombach, 1994), this goal is formalized as follows:

Analyze UXON

With the purpose of verifying if it is a suitable solution for evaluating Compomus UX[6]

Regarding its usefulness and feasibility

From the point of view of UX evaluators

In the context of UX evaluation by using interaction logging information.

In order to analyze the results, the following *indicators* were considered: usefulness and feasibility. The former was evaluated based on the perceptions of the UX evaluators of how much UXON helped them evaluate UX. The latter was evaluated based on the perceptions of the UX evaluators of how much ease and feasible they considered using UXON. Benefits and drawbacks pointed by the participants were also considered to indicate if UXON is useful and feasible.

The ***instruments*** used in the study consisted of: (i) a consent form to participate in the study, which aims to safeguard the participants' rights regarding the study and its results; (ii) a form to characterize the participants' profile, which aims to obtain information about the participants' knowledge of and experience in HCI evaluation; and (iii) a questionnaire that allows participants to record their perception after using UXON. The forms were prepared using Google Forms and are presented in Appendix A of this thesis.

The ***participants*** were the USES members (UX evaluators) who had previously evaluated Compomus UX without a specific supporting system (they extracted data from the interaction logs, imputed them in electronic spreadsheets and calculated the metrics). Thus, the participants had knowledge of how to evaluate Compomus UX and would be able to compare UXON with the previous solution they have used.

[6] The term UX, when referring to Compomus' UX throughout the text, refers to the immersive experience provided

The ***procedure*** adopted in the study consisted in making a brief presentation about UXON to USES members and making UXON available for use for around 30 days. After that, the three UXON users were invited via email to answer a questionnaire. After all of them accepting the invitation, the questionnaire was made available for 15 days. The provided answers were, thus, analyzed according to the study goal.

The questionnaire included 12 objective questions, whose possible answers are based on Likert scale. For each of them, the participants were asked to justify their answers. There was also a subjective question in which the participants could provide general improvement suggestions to UXON. The questionnaire was organized into three sections, namely: UXON Usefulness (7 questions); UXON Feasibility (4 questions); and, UXON General Evaluation and Suggestions (4 questions). A fragment of the questionnaire is presented in Figure 7.14. The complete form is presented in Appendix A.

7.2.5.1.2 Study Execution and Results

The participants of the study were three USES members. Participants P1 and P2 are Ph.D students who declared to have, respectively, high and medium theoretical knowledge of and practical experience in UX Evaluation. P3 is an senior researcher who had a Ph.D degree and declared to have high theoretical knowledge of and practical experience in UX Evaluation.

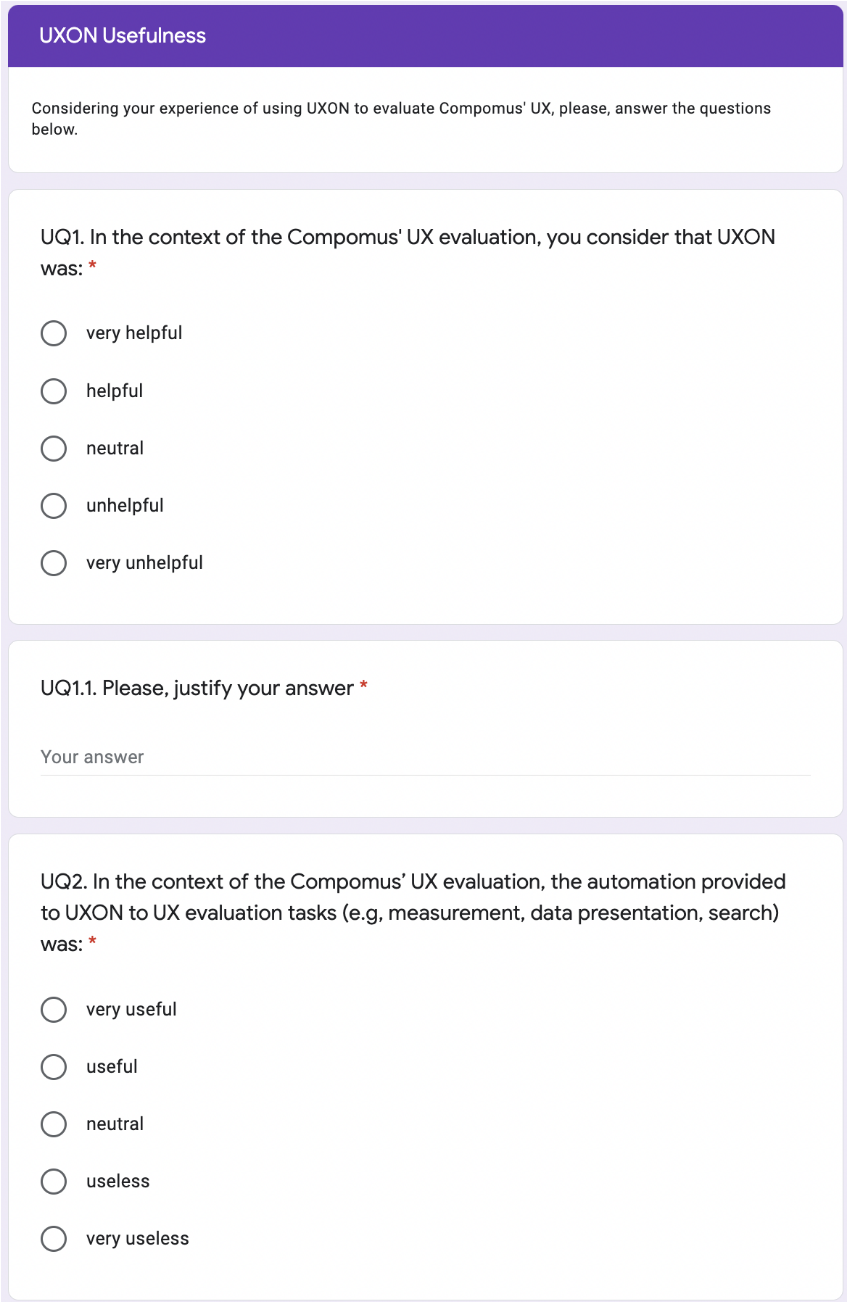
Following the planned procedure, the participants used UXON for about 30 days to evaluate Compomus UX considering data from several sessions. After that, they answered the questionnaire. Next, we summarize the main results obtained from the participants answers and comments.

UXON Usefulness

All the participants considered UXON *very helpful* to *evaluate Compomus UX*. According to P1, UXON plays an important role in supporting the analysis of UX data and that it automates a process that could take longer. P2 pointed out that the system is capable of plotting graphs from very large log files, which is challenging even for those who have some affinity with data processing systems. P3, in turn, emphasized that UXON offers useful metrics and graphs to analyze how Compomus users interacted and engaged.

All participants also considered the *automation provided to UXON* to UX evaluation tasks (e.g, measurement, data presentation, search) *very useful*. P1 stressed that data presentation is excellent, avoids more extensive analysis work and allows analyzing individual data, which previously required a lot of effort. P2 stated that UXON reduced the time to obtain the desired information. P3, in turn, pointed that UXON allows visualizing information in a simple way and that graphs and statistics are easy to analyze.

When asked to compare *data presentation* provided by UXON to the one provided by



UXON Usefulness

Considering your experience of using UXON to evaluate Compomus' UX, please, answer the questions below.

UQ1. In the context of the Compomus' UX evaluation, you consider that UXON was: *

- very helpful
- helpful
- neutral
- unhelpful
- very unhelpful

UQ1.1. Please, justify your answer *

Your answer

UQ2. In the context of the Compomus' UX evaluation, the automation provided to UXON to UX evaluation tasks (e.g, measurement, data presentation, search) was: *

- very useful
- useful
- neutral
- useless
- very useless

Figure 7.14 – Fragment of the questionnaire.

the previous used solution, two participants (P2 and P3) declared that UXON *improved* data presentation and one (P1) declared that UXON *strongly improved* it. P1 pointed out that prior to UXON there was no standard solution, and that this was a problem. P3 added that they used Google Collaboratory and that everything was very rudimentary, and required a lot of effort to calculate the metrics. P2 highlighted that UXON provides dynamic and editable graphics that help demonstrate more personalized graphics. P3 added that UXON's graphs facilitate the visualization of data and to compare the interaction between the participants in a visual way.

When asked to compare *the quality of data analysis* (based on the collected data) and

through the use of Compomus during a music composition session.

reached conclusions provided by UXON to the one provided by the previous used solution, two participants (P1 and P3) declared that UXON *strongly improved* the quality of data analysis and reached conclusions and one (P2) declared that UXON *improved* them. P2 pointed out that UXON supported new points of view from the data and P3 said that UXON facilitated and simplified data analysis, mainly by presenting graphical representations of the interactions.

All the participants considered UXON *neutral* regarding the improvements in the Compomus' UI/UX. All participants mentioned that it may be difficult to identify improvement opportunities based on the available log data.

When asked to compare the *time and effort spent* to evaluate Compomus' UX, using UXON and the previous solution, all participants considered that UXON *strongly decreased* them. P1 and P2 said that using UXON saved time and effort. P3 emphasized that it was much easier to perform the analysis with UXON and that previously, the analysis was performed manually, through scripts, and the cost of developing and testing these scripts is very high. As a drawback, P1 emphasized that due to the interactive graphics, there is a delay in data processing and that at this stage, a more visible loading message would fit for the evaluator to know that the data is being processed.

All participants considered that UXON *strongly supports data analysis & results* (e.g., data visualization, reaching conclusions) and *does not support* planning (e.g., definition of the metrics to be used) activities during an UX evaluation. Data collection (e.g., data capture, formatting and storage) activity was considered to be *strongly supported* by one (P3) participant and *not supported* by two (P1 and P2). P1 and P2 justified that planning and data collection activities are not supported, as UXON was not built for that purpose. P3 highlighted that UXON does not help in planning new metrics, as the metrics used are fixed. P1 stressed that UXON is extremely important for data analysis, P2 added that UXON helped in analyzing the data and reaching conclusions. P3 added that for data collection and analysis, UXON helps a lot.

UXON Feasibility

All participants found UXON *easy to use*. P2 justified his answer by saying that some graph customizations are not easy to perceive.

Regarding the *terminology* used in UXON, two participants (P1 and P3) considered it *very consistent* with the application domain (UX evaluation) and one (P2) found it *consistent*. P2 said that it is consistent only for those who are familiar with UX.

When asked if they would *use UXON again*, all participants answered *yes*. P1 explained that as an evaluator, UXON exponentially facilitated his job. P2 stated that UXON reduces work and time. P3 emphasized that they will use UXON whenever he needs to evaluate music composition sessions data and Compomus UX.

Concerning *recommending UXON for other people*, P1 and P2 answered that they *would*, while P3 said that he *would not*, because UXON is specific to evaluate Compomus UX and other people may not be interest in evaluating that application.

General Aspects

When asked about *extending UXON to support UX evaluation of other applications* whose evaluation is based on interaction logging data, all participants considered it would be *very useful*. P1 emphasized that evaluating UX through log file's data is a big challenge, therefore, the role of UXON is fundamental. P2 added that it would be very valid, as the analysis of large volumes of data requires knowledge of several programming languages and that the data can be analyzed more easily by a ready-made system. P3 emphasized that UXON would help evaluators to work with the log files of other applications in an easy way, providing useful information and easy analysis.

Regarding the main *advantages* of using UXON, the participants listed: support in the analysis process; visual analysis of data; analysis by participant and overall experience; individual analysis of UX metrics for the immersive experience, allowing conclusions regarding the experience; practicality; ease of use; simplicity in the process of loading log files; graphical presentations that make it possible to have an overview of the interaction; diversity of forms of data organization (different graphics); and possibility of performing new queries.

As for the main *disadvantages* of using UXON, the participants cited: data processing time; graphic quality of the evaluation report; prepare log file as input without knowing the data format of the database expected by the system; and UXON only works for Compomus.

Finally, when asked about *suggestions* to UXON improvement, the participants answered: improve data processing time; make some graphs more explanatory; improve the layout of the data/graphics presentation; provide a documentation that informs which fields are required and expected by the system for the evaluator to prepare its log file; and, offer help in creating new queries.

7.2.5.1.3 Discussion

In this section, we make some discussions about the results presented in the previous section in terms of the indicators established in the study planning.

Concerning *usefulness*, we observed that, in general, all participants had the same or a very close perception. They agreed that UXON is very helpful, useful, automates UX evaluation tasks, improves the quality of data presentation and data analysis, and strongly decreases time and effort spent to evaluate Compomus UX. They also agreed that the use of UXON was neutral to help identify improvements, when compared to the previous solution. We believe that this is due to the metrics currently available, which may be not enough to provide information to suggest improvements in Compomus UI/UX. There is an ongoing doctorate research investigating UX/UI measures at USES, thus new metrics can be added to UXON in the future, to provide a more comprehensive view of UX/UI aspects and contribute more effectively to identify improvement opportunities.

Regarding the UX evaluation process, the participants agreed that UXON strongly supports data analysis and does not support evaluation planning. The participants disagreed on the support to data collection. We believe that this is due to a different understanding of data collection scope in the Compomus – UXON context. P1 and P2 (Ph.D students) considered that data collection is performed by Compomus, since it records interaction data in the log file. On the other hand, P3 (the most experienced participant) considered that data collection is supported by the ETL process, which captures data from the log file, stores it in UXON data base and use it to calculate the metrics values.

Regarding *feasibility*, the participants agreed that UXON is easy to use. Moreover, they would use UXON again and most of them would recommend UXON to other people. P3 told that he would not recommend it because UXON is specific to evaluate Compomus UX and other people may not be interested in that. Although this does not exactly represent a limitation, as the system was developed to specifically support Compomus, it points to the need to evolve UXON to handle data from other applications.

As for UXON terminology, the participants agreed that it is consistent with the application domain (UX evaluation) but P2 mentioned that the terminology is only suitable for those who are familiar with the UX domain. We understand the participant concern. However, considering that the terminology is for a specific domain, it should be consistent with that domain. Thus, it is expected that people not familiar with that domain may use other terms and not know terms often used in the domain. Thus, we believe that the terminology meets the main stakeholders needs, and its suitability for other audiences would require further investigation.

As for advantages of using UXON, participants reinforced automation, data representation, simplicity, ease of use, support to data analysis and decrease of time/effort.

Concerning disadvantages, they highlighted data processing time, evaluation report graphic's quality, and the fact that UXON only works for Compomus. These limitations will be considered in future improvements. We agree that we need to reduce processing time and show the processing progress to the user until it is finished. Also, usability and interactive graphics can be improved and UXON can be extended to aid in UX evaluation of other applications. Regarding "prepare log file", as Compomus generates the log file in text format, the only preparation required is to transcribe this file into comma-separated values (CSV) format, which the UXON receives as input. This is a simple procedure, performed even by non-specialists. In other words, the system does not require any other preparation of the Compomus log file other than transcribing it to CSV format, nor does it require the UX expert to understand the format of the data in the system's database.

In summary, based on the participants perceptions, UXON was considered as a *promising system*, *very helpful*, *useful* and *easy to use*. Moreover, there are more advantages of using UXON in UX evaluation than disadvantages. Hence, we can conclude that there is indication that UXON is useful and its use feasible.

7.2.5.1.4 Threats to Validity

As any study, this study has some limitations that may have threatened the validity of its results. Thus, these limitations must be considered together with the results. The threats related to this study have been divided into categories, as proposed by (Runeson et al., 2012), and are presented below.

Internal Validity: is defined as the ability of a new study to repeat the behavior of the current study with the same participants and objects with which it was carried out. One of the main threats to internal validity is communication and information sharing among study participants. To minimize this threat, the researcher sent the questionnaire to the participants by email and the participants were allowed to answer it when and where they wanted. They were asked to answer the questionnaire by themselves, without communication with the other participants. Another threat to be considered is that the study was carried remotely, thus the participants may have performed other tasks parallel to the study. If we could guarantee that they were exclusively focused on the study, maybe the study results could have been different.

External Validity: this threat is related to the ability to repeat the same behavior with groups different from the one that participated in the study. The main threat in this category refers to the small number of participants. However, considering that they are a representative sample of the population because, at this moment, there are few people that evaluate Compomus UX, we believe that this threat is minimized. Another threat also related to the participants concerns the fact that they were involved in UXON development. Since USES were the 'client' of the proposed solution, they were the stakeholders that presented us the problem and provided information so that it was possible to propose the solution. During UXON development, they received an MVP (Minimum Viable Product) and evaluated it. Based on the provided feedback, we developed the current version of UXON.

Construct validity: refers to the relationship between the instruments and the study participants and the theory being proved. For this category, the main threat identified concerns the possibility of misuse UXON functionalities due to the lack of documentation. To minimize this threat, before the participants use UXON, the researcher performed a presentation introducing it to P1 and P2 (P3 was not present). There is also the threat of the participants have misunderstood questions contained in the questionnaire. To address this threat, the researcher was available to answer questions and support the participants. The questions can also be a threat to the results. Some of them can lead to confirmation bias. We minimized this threat by asking the participants to justify their answers, so that they could reflect about the given answers instead of only answer positively or negatively.

Reliability Validity: concerns the extent to which the data collected and analyzes performed in the study depend on the researcher who conducted it. The doctoral candidate together with her advisor conducted the analysis of the data provided by the participants in the forms. Thus, the interpretation performed is dependent on the candidate and her advisor. However, considering that the study involved only three participants and that the answers given were very clear, possibly the results obtained would be similar even if another researcher had analyzed the data provided by the participants. However, the threat of different interpretations and, consequently, different results, is not excluded.

Considering these threats, the study results cannot be generalized and must be understood as preliminary evidence that UXON is useful and feasible to support Compomus UX evaluation.

7.2.5.2 UXON Evaluation – Developer Perspective

This section presents an interview carried out with the UXON developer users aiming at to obtain feedback about the use of HCI-ON in the development of systems to support the solution of HCI-related problems. Section 7.2.5.2.1 presents the interview planning; Section 7.2.5.2.2 addresses its execution and main results; in Section 7.2.5.2.3, we briefly discuss the results; and in Section 7.2.5.2.4 we present some limitations.

7.2.5.2.1 Study Planning

The interview **goal** was to investigate, from the developer point of view, whether the use of (an extract of) HCI-ON helps the development of a system to support solving HCI-related problems. Aligned to this goal, we defined two main **questions**:

(Q1) How does the use of (an extract of) HCI-ON help in the development of systems to support the solution of HCI-related problems?

(Q2) What are the benefits and difficulties of using (an extract of) HCI-ON in system development to address HCI-related problems?

The **instruments** used in the study consisted of: (i) a consent form; (ii) a form to characterize the participant profile; and (iii) a questionnaire for the interviewer to follow during the interview. The forms were prepared in Google Forms and are presented in Appendix B of this thesis.

The **procedure** adopted in the study consisted of a face-to-face approach and semi-structured interview. In the face-to-face approach an interviewer asks the questions in the presence of the respondent, and also completes the questionnaire (Robson; McCartan, 2016). In the semi-structured interview, the interviewer has an interview guide that serves as a checklist of topics to be covered and order for the questions. Based on the flow of the interview, the order can be substantially modified and additional unplanned questions can be asked to follow up on what the interviewee says (Robson; McCartan, 2016).

The **participant** was the UXON developer, who developed UXON as part of her monography. It is worth noting that the developer was co-supervised by the author of this thesis.

In the interview questionnaire Q1 and Q2 were detailed in more specific questions to be answered. They are listed bellow.

- Q1.1. How did you use HCI-ON extract to develop UXON?
- Q1.2. In which stages of UXON development (analysis, design,...) was the ontology most useful? Why?
- Q1.3. In which stages did the use of ontology not help? Why?
- Q1.4. Do you consider that the HCI-ON extract helped you to have a better understanding of the domain addressed in the developed application (UXON)?
- Q1.5. Do you consider that the HCI-ON extract used in the development of UXON was able to cover the HCI domain treated in the application?
- Q1.6. How did the semantics provided by the conceptualization (from the extract) of HCI-ON (e.g., concepts and their descriptions/meanings, relationships between concepts) help in the development of UXON? Why?
- Q2.1. What benefits have you noticed when using HCI-ON extract in UXON development?
- Q2.2. What difficulties did you face when using HCI-ON extract?
- Q2.3. Was this the first time you developed an ontology-based system? Briefly describe your experience.
- Q2.4. Would you use ontologies again to develop another system? Why?

7.2.5.2.2 Study Execution and Results

The participant of the study was the UXON developer who has undergraduate degree in Computer Science and declared to have high theoretical and practical knowledge of systems development and medium knowledge of ontologies and ontology-based system development.

During the interview, the interviewer followed the questionnaire. One question was added during the interview. With the consent of the interviewee, the interview was recorded.

Following the planned procedure, the interview was conducted face-to-face. After presenting the interview goal, the interviewer started the interview following the questionnaire. During the interview, the interviewer changed the order of some questions. One new question raised during the interview (How do you contrast ontology-based and non-ontology-based software development against the semantics provided by the former?). Although the interviewer had a questionnaire to follow, she rephrased some questions, presented examples and interacted with the interviewee to improve her understanding and collecting feedback.

Next, we summarized the main results obtained from the participant answers and comments.

When asked about the use of (an extract of) HCI-ON and if it helped in the development of UXON, the participant reported that she used the ontology (i.e., the HCI-ON extract) to create the conceptual model of UXON and understand *ontoUXON* (Q1.1). According to her, the ontology was more helpful at the Analysis stage, supporting the understanding of the domain (concepts and relationships) for requirements gathering, thinking about and elaborating the requirements, and the things that the system should display (Q1.2). On the other hand, she said that the ontology was not much useful in the Design stage, specifically in the architectural design, e.g., in how to organize the components. She reported that although the ontology helped in the Implementation stage, she had difficulties with the adopted technology and did not have time to study it (Q1.3). She stated that the ontology helped better understand the domain (Q1.4-5). She said that the semantics provided by the ontology helped in the development process (for example, it helped to create the triplestore, assign semantics to concepts and relationships, and create of SPARQLs queries) (Q1.6). Concerning the benefits of using HCI-ON, she cited as the main benefit its support to understanding the domain and added that this considerably reduced the learning curve when compared to a non-ontology-based system. In her words, in non-ontology based development it would take more time and effort to understand the domain (Q2.1). Regarding the difficulties, she said that she did not have difficulties related to ontologies and their manipulation, but to the use of the technologies necessary to implement them (Q2.2). She declared that this it was the first time she developed an ontology-based system (Q2.3) and that, based on this experience, she would use ontologies to develop other systems (Q2.4). She briefly explained her experience in developing the UXON as positive and reported that while she found it interesting and would use ontologies again, in her current job, the work is focused on non-ontology-based systems. She added that she would certainly use the ODD approach (domain understanding in development time) to generate the conceptual model. Regarding the OBA approach, she reported that she perceived a lack of available knowledge. Therefore, she said that she would not use the OBA approach in future developments.

7.2.5.2.3 Discussion

In this section, we present a discussion about the results presented in the previous section in terms of the questions defined on the study planning.

Regarding Q1, the results from the study indicated that use of (an extract of) HCI-ON helped in the development of an application to support the solution of HCI-related problems. According to the UXON developer, the HCI-ON extract provided great support to the Analysis stage of UXON development. Furthermore, HCI-ON extract helped to gain a better understanding and coverage of the HCI domain addressed in the application, and its semantics certainly helped in the UXON's development. As a drawback, the participant stressed that there was a lack of support in the Design and Implementation stage. As for the Implementation, the limitation was related to the used technologies instead of the ontology.

Concerning Q2, the developer feedback indicated that the main benefits of using (an extract of) HCI-ON are the ease of understanding the domain and shortening the learning curve when compared to non-ontology-based software development. Furthermore, the developer suggested that the ODD approach may be more suitable for novices in ontology-oriented software development or with little experience in ontologies, while OBA application may require more expertise. As reported by the participant, her greatest difficulty was not in the use of the HCI-ON extract itself, but in the used technologies. As she reported, this difficulty may be due to lack of knowledge and time to study. Moreover, the use of operational ontology in software coding is already a challenge, especially for non-experienced people (which was the case of the participant). When this is combined with the use of technology not known for the developer, we believe that these difficulties can be increased.

The overall results of the interview indicated that the use of (an extract of) HCI-ON helped in UXON development (from its developer's point of view). Despite its greater contribution had been in the development time (ODD approach), it was not possible to properly evaluate how HCI-ON contributed in the run-time because the developer had difficulties to use the adopted technologies (e.g., Flask, Python), which may have prevent her from perceiving the actual impact of using HCI-ON at run-time.

7.2.5.2.4 Limitations of the Study

The results discussed in the previous section should be considered together with some limitations involved in the study. The main limitation regards the participation of this doctoral candidate in UXON development. As we explained before, the UXON developer carried out UXON development in her undergraduate project, which was co-supervised by this doctoral candidate. This may have influenced the developer perception of using HCI-ON. Moreover, this may also have influence on the interviewee answers.

To minimize the influence of the relation between interviewee and interviewer during the interview, the interviewer (this doctoral candidate) followed some recommended procedures: she listened more than she spoke; posed questions in a straightforward, clear and non-threatening way; and tried to get interviewee to talk freely and openly. Even so, it is not possible to eliminate biases.

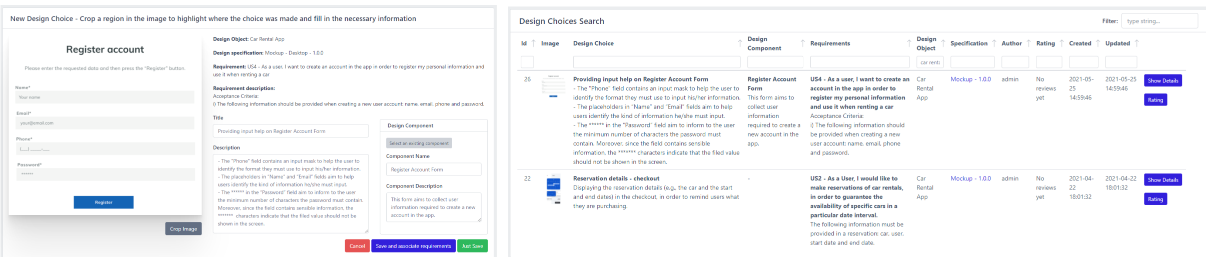
Another limitation concerns the developer profile. She is a beginner in ontology-oriented software development. The fact that she was the only developer of UXON is also a limitation because we could not get feedback from other people. Some limitations inherent in interviews in general also apply to this study. First, interviewing is time-consuming. It can make the interviewee tired, influencing their answers. The interview was very straightforward and lasted 38 minutes. It was recorded, so the interviewer did not need to spend time writing the answers. Second, the participant may misunderstand some questions. To avoid this, the interviewer exemplified some questions and/or redrafted the questions in order to facilitate understanding. Moreover, she changed the order of the questions to better adapt to the flow of the interview. Last, some questions can lead to confirmation bias. In such cases, the interviewer asked the participant to justify her answers.

Considering the study limitations, the results are not conclusive and should be considered as preliminary evidence that the use of (an extract of) HCI-ON helps develop systems to address HCI-related problems.

7.3 Knowledge Tool for Interaction Design - KTID

KTID, a tool that aims to support the capture and sharing of useful knowledge to assist in HCI design, was developed in the context of a master research (Castro, 2021) and an undergraduate project (Sessa, 2021) and uses a fragment of HCI-ON, which includes concepts from HCI Design Ontology (HCIDO) (Costa et al., 2020; Castro, 2021) (shown in Figure 7.1c), HCIO and some ontologies of SEON. The development of KTID followed the ODD approach, and the HCI-ON fragment was used as a reference model at the development time to structure the knowledge tool and its relational database.

KTID[7] supports knowledge management (capture, representation, storage, retrieval, use and evaluation) to aid in HCI design of interactive systems. It allows HCI designers to annotate structured information about design choices into design artifacts and share it with other HCI design stakeholders (Castro, 2021; Castro et al., 2021; Castro et al., 2022). Figure 7.15 shows screenshots of two KTID's pages/features. Figure 7.15a illustrates the recording of a design choice while Figure 7.15b shows the KTID page used to search for design choices.



(a) Describing a design choice. (b) Searching for design choices recorded.

Figure 7.15 – Screenshots of KTID (Castro et al., 2021).

KTID was evaluated by two HCI designers and it was considered useful and feasible

[7] KTID is available at <https://dev.nemo.inf.ufes.br/ktid/>.

(Castro et al., 2021; Castro, 2021). Although not developed in this work, like UXON, KTID is an example of HCI-ON application and serves to demonstrate that extracts of HCI-ON can be used to support system development.

7.4 Social Network with Ontology-based adaPtive Interface - SNOPI

SNOPI was developed in the context of an ongoing doctoral research and an undergraduate project (Scalser, 2022). It uses a fragment of HCI-ON including concepts from HCIO, User Characterization Ontology (UCO) and UI Types and Elements Ontology (UIT&EO) (shown in Figure 7.1d). These ontologies are being developed in the context of the aforementioned doctoral research. The already built part (used in SNOPI) is presented in the work by (Scalser, 2022). Like KTID, SNOPI served as a case to evaluate and demonstrate the use of HCI-ON.

The development of SNOPI followed the ODD and OBA approaches, and the HCI-ON fragment was used as a reference model at dev-time (conceptual model) to structure the social network and its relational database, and as computational artifact (ontoSNOPI) at run-time to support the user interface adaptations through reasoning.

ontoSNOPI[8] is the HCI-ON fragment (conceptual level, Figure 7.1d) transcribed into a language that allows machine-reading (implementation level). ontoSNOPI was built using OWL, and encapsulates information (rules), through axioms, on how to adapt the interface according to the user's profile (degrees of color blindness and low vision). The axioms (rules) were built considering the recommendations of the W3C Accessibility[9] and the Web Content Accessibility Guidelines (WCAG)[10].

SNOPI[11] focuses on adaptations of the user interface of an academic social network for users with different degrees of color blindness and low vision (Figure 7.16). When accessing the system, the user is asked to answer a questionnaire, which records data from the user (*UI* and *application layers*, respectively). At run-time, the recorded data is instantiated in the ontoSNOPI (User Profile classes), and inferences (reasoning) through the axioms are performed on these instances. Thus, the user profile is classified into W3C recommended accessibility adaptations for his UI (*semantic layer*). An algorithm identifies the recommended adaptations, records in the database and then proceeds to adapt the UI (*application*, *data*, and *UI layers* respectively). This procedure is performed at the first time the user accesses the system. In the next accesses, the algorithm checks if there are profile and adaptations records for the user in the database tables and presents the corresponding UI (*application*, *data*, and *UI layers* respectively). The user can also reset their profile at any time and answer the questionnaire again.

---

[8] ontoSNOPI is available at <https://dev.nemo.inf.ufes.br/hcion/ontoSNOPI.owl#>.

[9] <https://www.w3.org/standards/webdesign/accessibility>

[10] <https://www.w3.org/TR/WCAG/>

[11] SNOPI is available at <https://dev.nemo.inf.ufes.br/snopi/>.

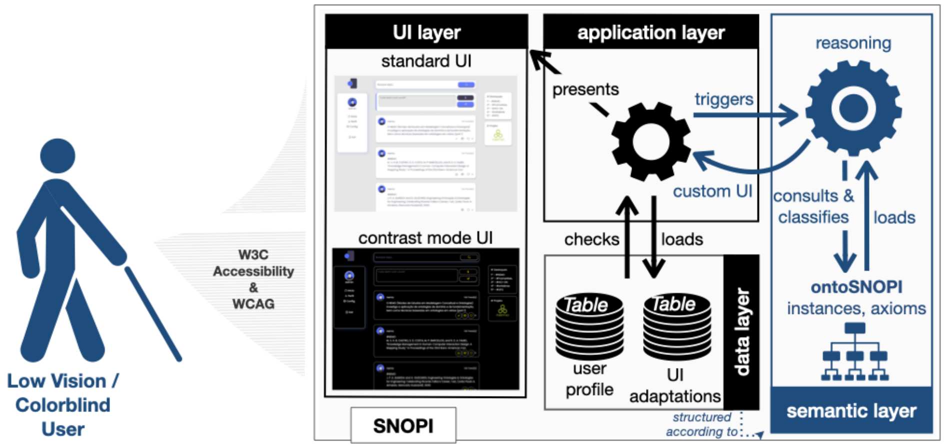
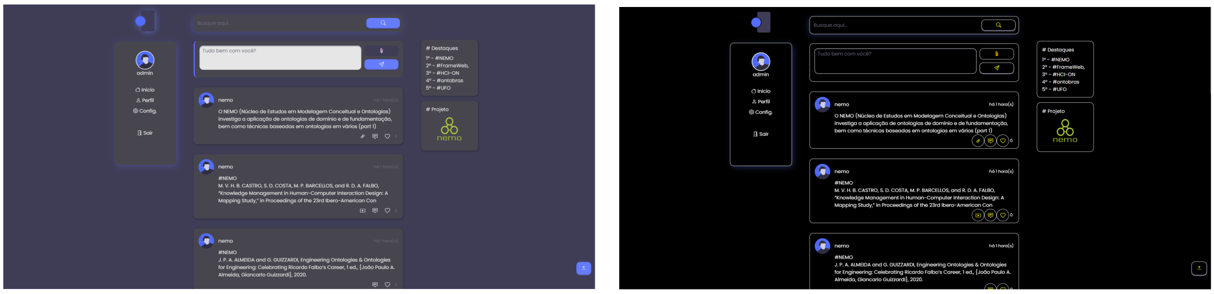


Figure 7.16 – SNOPI overview.

Figure 7.17a, illustrates the adaptation of the interface to the dark mode (low light emission). This adaptation is suitable for colorblind or light-sensitive user. Figure 7.17b, presents the adaptation of the interface to the high contrast mode (increases contrast, low visual interference, increases the focus to the components). This adaptation is suitable for contrast sensitive user.



(a) feed page adapted for dark mode. (b) feed page adapted for high contrast mode.

Figure 7.17 – Screenshots of SNOPI (Scalser, 2022).

7.5 Final Considerations of the Chapter

This chapter presented some applications of HCI-ON in the development of ontology-based systems. In this context, three applications were developed, namely: UXON: User eXperience Ontology Network-based system; KTID, Knowledge Tool for Interaction Design; and SNOPI, Social Network with Ontology-based adaPtive Interface.

UXON development served as a proof of concept that showed that the use of HCI-ON in the development of systems to address HCI-related problems is feasible.

As for main benefits of using HCI-ON in the development of UXON, we point out: (i) the development of UXON conceptual model based on a general conceptualization of HCI evaluation, rather than on a particular application, allowed us to have a more structured view and propose a suitable solution for the problem; (ii) HCI-ON conceptualization allowed spending less effort in the conceptual modeling of UXON because it was just necessary to make a few adaptations in the ontology model; and (iii) less effort and time was spent in coding and building the database.

KTID and SNOPI were developed in the context of other students' research projects and served as cases for demonstrate the use of HCI-ON for developing applications by third parties, since HCI-ON was used by others to develop the applications. However, in both cases, people involved in the development of HCI-ON networked ontologies participated in the system development.

In summary, the use of HCI-ON extracts as a basis to develop HCI systems, both at dev and run-time, served as proof of concept that showed that it is feasible to use HCI-ON to develop HCI solutions. As a proof of concept, the results only indicate that using HCI-ON is viable, and it does not mean that HCI-ON is useful in practical settings. Therefore, studies about the use of HCI-ON by third parties than the researchers involved in HCI-ON development still need to be conducted.

8 Final Considerations

This final chapter presents the final remarks and conclusions regarding the research presented in this thesis. In Section 8.1, a brief summary points out the main aspects presented along the text. Section 8.2 describes the produced contributions, relating them to the proposed objectives and to the published papers. Section 8.3 discusses the limitations of the work. Finally, Section 8.4 presents recommendations for future improvements and research.

8.1 Summary of the Research

This research addresses a knowledge framework of the HCI domain to support HCI's knowledge-related problems and semantic interoperability. The framework is defined through an ontology network, whose architecture and evolution mechanisms enable knowledge structuring, reuse, and evolution.

This research was motivated since, over almost three decades, the aforementioned HCI's problems have been partially addressed (Costa; Barcellos; Falbo, 2021). On the other hand, we were also motivated by our research group's knowledge and experience of decades of working with ontologies, which are appropriate instruments to solve the aforementioned problems.

Through a Systematic Review of the Literature (Costa; Barcellos; Falbo, 2021) we have identified and analyzed studies addressing how ontologies have been used in the HCI domain, producing an overview of the current research status. Despite many works that have produced contributions to the field, ontological treatment is still in the initial stage. We also identified research gaps on how the proposed solutions deal with the ontology underlying ontology engineering and limited coverage of the HCI domain underlying the diversity of knowledge involved in the HCI domain.

Due to the diversity of HCI knowledge and based on experience with ontologies, we identified the need for a more robust solution. The definition of the HCI knowledge framework was strongly inspired by SEON (Ruy et al., 2016), a recognized and successful work in ontology networks for Software Engineering developed in our research group. As predicted in the thesis by Ruy (2017), third parties can reuse the SEON framework. In the case of HCI-ON, we used some elements of the SEON framework, making some changes and extensions.

HCI-ON has a specific purpose within this thesis. It serves as an HCI integrated knowledge reference purpose to be applied in knowledge-related and semantic interoperability problems. We argue that by organizing HCI ontologies in a network (HCI-ON), knowledge is better structured, enabling us to reach a more comprehensive conceptualization where ontologies reuse concepts one from another, keeping consistency in shared concepts and decreasing overlap problems. By organizing HCI ontologies in an ON, when ontologies are needed in scenarios spanning different HCI sub-domains, instead of spending effort to integrate several ontologies to cover the portion of the domain of interest, one can just extract the ON portion to be used. Thus, we advocate that the effort needed to develop and organize ontologies in a network is compensated when portions of the network are applied which requires less effort to solve HCI problems.

Thus, in this thesis, we further explore three main aspects. First, ontologies depend on a consistent domain knowledge representation, offering a solution-independent conceptualization to support knowledge-related and semantic interoperability solutions. Second, the conceptualization of vast domains, such as HCI, requires modularization to be viable and easy to use and enable procedures that support integration and growth. Third, the development of ontologies must follow methods, adopt an architecture, envision quality characteristics, and ontology reuse encouraged by good practices in ontology engineering. Aimed to fulfill the domain knowledge representation and modularization needs, we have created HCI-ON. It is an Ontology Network for the HCI domain composed of a foundational ontology, a core ontology describing the human-computer interaction phenomenon, and well-founded and aligned domain ontologies covering diverse HCI subdomains.

Being an ontology network, HCI-ON can constantly evolve. Therefore, the version presented in this thesis is not a final artifact or a complete HCI ontology. Contrariwise, it is an effort to provide initial content for a long-term evolution work, aiming at being a useful and consistent source of HCI domain knowledge. We explored and adapted SEON's mechanisms to define mechanisms to support HCI-ON evolution, consisting of four different ways that ontology engineers can follow to add ontologies to the network and evolve its conceptualization. We believe they are important elements and contribute to reusing networked ontologies or their fragments. Moreover, networked ontologies are well-suited to combine information from various sources (HCI sub-domains) and infer new facts based on this. Also, the ON flexibility allows an easy extension of existing ontologies, thus fostering the reuse of existing work. Currently, HCI-ON comprises eight ontologies (some under construction) and more than 90 concepts regarding five HCI subdomains. HCI-ON conceptualization has been successfully applied in distinct systems development to support solving HCI problems. In these initiatives, HCI-ON has been pointed out as an important artifact to support the understanding of the application domain.

For evaluating the feasibility of the evolutionary mechanisms, this doctoral candidate successfully used them for HCIO and HCIEO development and literature ontologies alignment. Moreover, third parties also used them to HCIDO (Costa et al., 2020; Castro, 2021), UIT&EO, and UCO development (these last two are under development).

Aiming to evaluate the use of HCI-ON to solve HCI-related problems, we developed UXON (Manso, 2022). The users' perspective evaluation results indicated that the system is useful. The developer's perspective evaluation findings showed that it is feasible to use HCI-ON to solve problems in the HCI domain. HCI-ON was also used in other works to develop other systems (Castro et al., 2021; Sessa, 2021; Scalser, 2022) that address HCI-related problems.

These applications provided us with a practical view of using HCI-ON in ontology-oriented software development, helping us identify and understand some research limitations and perspectives for future work.

8.2 Research Contributions

This work involves two areas, Human-Computer Interaction (HCI) and Ontology Engineering (OE). As presented in Chapter 1, the general objective of this work was to *propose a knowledge framework of the HCI domain, which provides a comprehensive conceptualization of that domain, favors knowledge growth, reuse, and integration, and supports knowledge-based and interoperability solutions*. We derived this objective from the main problem addressed in this work: the lack of a comprehensive conceptualization of the HCI domain. This issue affects the HCI area and its vast and multifaceted community (i.e., researchers, teachers, and professionals), wherein each one adopts different definitions. Some research on the HCI domain has applied ontologies in HCI knowledge to obtain knowledge and semantic interoperability solutions.

This research contributes to both areas, HCI and OE. On the one hand, we put the OE body of knowledge into practice and occasionally produced knowledge. On the other hand, HCI is the area that represents the research ends. Therefore, we understand this research as an EO application to solve HCI problems. HCI-ON is the main contribution of this research. Efforts employed in obtaining the HCI-ON are also part of our contributions, namely:

- **Human-Computer Interaction Ontology Network (HCI-ON):** HCI-ON was conceived as our HCI reference knowledge framework to be used as a semantic reference in semantic interoperability and knowledge-related efforts. The current version of HCI-ON, including HCI-ON architecture, evolution mechanisms, core ontology, and domain ontologies body are relevant achievements for promoting that purpose. HCI-ON is a Human-Computer Interaction Ontology Network designed seeking to establish a structure for supporting ontologies representing different HCI aspects, levels, and sub-domains; providing effective support for integrating and aligning domain ontologies; being applied for solving semantic interoperability and knowledge-related problems in HCI. We have applied the ideas of Ontology Networks (Suárez-Figueroa et al., 2012; Ruy et al., 2016) and developed HCI-ON, starting with its core ontology (HCIO) and, then, developing domain ontologies in the context of this and other works. HCI-ON is a way to integrate ontologies in an evolving framework, assuring the network premises, and maximizing their application in diverse research initiatives. HCI-ON was introduced in the paper entitled "*Towards an Ontology Network on Human-Computer Interaction*", published in the proceedings of the 39th International Conference on Conceptual Modeling (Costa et al., 2020). Its current version was presented in detail in Chapter 4, and its specification is available at <dev.nemo.inf.ufes.br/hcion/>. Some results produced during HCI-ON development are contributions themselves:

- *HCI-ON's Architecture:* we extended the architecture proposed by (Ruy et al., 2016; Ruy, 2017) and the resulting architecture can be used in other ontology networks.
- *Mechanisms to support HCI-ON evolution:* the defined mechanisms are an initial guide to be followed by ontology engineers to maintain a consistent evolution of the network conceptualization and among its networked ontologies. They can also be used in the context of other ontology networks.
- *Body of Ontologies:* currently, HCI-ON has HCIO as core ontology and seven well-founded domain ontologies for different HCI subdomains, counting around 90 concepts (not counting UFO and counting UIT&EO and UCO that are in more advanced stages of development). All network concepts are grounded, defined, and contextualized relative to their networked ontologies. The body of knowledge can be applied in diverse contexts, as discussed in Section 8.4.
- *HCI-ON Specification:* the HCI-ON specification reuses and adapts the solution proposed in (Ruy, 2017) to make the ontology models and related information available on a website. As SEON's Specification, it enables easy access to the network contributing to the work visibility and reuse.

- **Human-Computer Interaction Ontology (HCIO):** a reference ontology about the Human-Computer Interaction phenomenon. Addresses what an interactive computer system is, its components, and its user interface; user actions are taken in the course of an HCI and the user motivation to start the interaction; and how an HCI happens. HCIO was presented in detail in Chapter 5. A paper presenting HCIO (entitled "A Core Ontology on the Human-Computer Interaction phenomenon") was published in Data & Knowledge Engineering journal (Costa et al., 2022).

- **HCI Evaluation Ontology (HCIEO):** a well-founded reference domain ontology about HCI evaluation. Its conceptualization addresses HCI evaluation, artifact produced, evaluation criteria, quality characteristics, user participation, as well as qualitative and quantitative evaluation. HCIEO was presented in detail in Chapter 6.

- **Secondary Study investigating HCI ontologies:** a study performed combining systematic literature review and systematic mapping, aiming to obtain an overview of existing HCI ontologies, their applications, and quality characteristics. The study provides information about which portions of the HCI domain have been supported by ontologies; types of ontology-based solutions that have been adopted to solve HCI problems; how HCI ontologies have been developed and evaluated and the quality characteristics they have presented.

35 ontologies were identified and analyzed. The study revealed a panorama of the topic, which can be valuable material for future HCI ontologies research efforts. In the context of this thesis, we highlight the findings and the research gaps. These points were essential for determining our main directions regarding our knowledge framework (HCI-ON) and its application in practice. The spreadsheets produced during the study can be found in (Costa; Barcellos; Falbo, 2020c). The study was presented in Chapter 3. A paper presenting the secondary study and its main results (entitled “Ontologies in Human-Computer Interaction: A systematic literature review”) was published in the Applied Ontology Journal (Costa; Barcellos; Falbo, 2021).

- **UXON:** The User eXperience Ontology Network-based system focuses on providing data analysis support to UX experts during the Compomus UX (immersive experience) evaluation. In collaboration with an undergraduate student, the system was developed (Manso, 2022) and is available for use. UX experts from USES used the system to support evaluating the UX of Compomus (Chapter 7).

Table 8.1 relates the presented contributions to the specific objectives of this thesis, showing that all the established objectives were met.

Table 8.1 – Contributions versus Specific Objectives.

General Objective	Specific Objective	Contribution
Propose a knowledge framework of the HCI domain, which provides a comprehensive conceptualization of that domain, favors knowledge growth, reuse and integration, and supports knowledge-based and interoperability solutions	**SO1.** Investigate the state of the art about ontologies in the HCI domain	A Secondary study on HCI Ontologies
	SO2. Establish an HCI reference knowledge framework	The Human-Computer Interaction Ontology Network
	SO3. Establish mechanisms to support knowledge access, creation, integration and evolution	Mechanisms to support HCI-ON evolution
	SO4. Apply the knowledge framework to solve knowledge-related and interoperability problems in the HCI domain	User eXperience Ontology Network-based system (UXON)

Finally, the results produced throughout this thesis confirm our research hypothesis: knowledge-related and semantic interoperability problems in the HCI domain can be addressed with a consensual, shared, and comprehensive conceptualization of HCI, represented by means of an ontology network.

8.3 Research Limitations

Like any research, this work has limitations. Throughout the chapters, we discussed some specific limitations. In this section we present some limitations regarding the work as a whole.

HCI-ON is the main contribution of this work. In HCI-ON context, we point out the following limitations:

- **Evolution Mechanisms:** we have provided a set of initial instructions on how to evolve HCI-ON and the proposed mechanisms were used exclusively by members of our research group. Adding an ontology to a network is a complex activity, requiring further research and external use/evaluation.
- **HCI-ON Coverage:** currently, HCI-ON covers aspects of seven different HCI subdomains. It is a good starting point, though, along the time, ontologies on other HCI disciplines should be developed and added to the network.
- **HCI-ON Evaluation:** Although our research group developed three systems using HCI-ON, which demonstrates HCI-ON feasibility of use, each of these systems requires the use of specific network extracts. Because of this, we did not use the entire network. Thus, we evaluated some concepts only through verification (using competence questions) and validation (with instantiations) techniques, and expert review. Still regarding HCI-ON evaluation, in this thesis, we explored HCI-ON use to support the development of systems, where extracts of HCI-ON were used as conceptual models and as operational ontologies. However, none of the solutions address interoperability problems. This, the requirement R9 and the acceptance criterion C3 were partially met.
- **Contributions to HCI-ON:** up until now, the network's development and growth have been done exclusively by our group. It is necessary to allow other researchers to contribute to HCI-ON.
- **UXON:** is a secondary contribution, produced to evaluate the use of HCI-ON. The main limitation related to UXON is that it only works for Compomus. This does not allow using the proposed solution in different scenarios, to evaluate different applications, which would enable a more robust evaluation of the use of HCI-ON.

8.4 Perspectives of Future Works

Below we present some directions and perspectives for future works related to this research.

- **Network Evolution**: the evolution mechanisms can further be explored. We have provided some guidelines for HCI-ON evolution, but it should offer more detailed guidance for ontology engineers, considering ontology grounding, the similarity between concepts, and integration scope.

- **HCI-ON Coverage**: HCI-ON's body of ontologies shall be increased, allowing the HCI domain better coverage. Ontologies covering other subdomains should be developed and added to HCI-ON (e.g., Design and Evaluation Process, Evaluation Methods, Context Awareness Systems, Psychological Computing).
- **HCIO:** the modeling of ontology concepts and relationships and its scope commit to the interaction phenomenon as a communication (dialogue) between the user and the system point of view. Therefore, it may not fit into situations that consider interaction from other points of view.
- **HCIEO:** despite every effort employed in its accomplishment in the future, it may be necessary to evolve it to cover new situations.

Along with UXON development, diverse ideas related to its **improvements** have been applied and some remained as future perspectives.

- **Domain Coverage**: as suggested by the survey findings, extending the system context to encompass the UX analysis of other interaction logging applications would be very useful as this type of analysis without the support of an analytical system is a major challenge.
- **Specific Improvements**: despite the recognition of their support and help, the result of the study also stressed some specific improvements: data processing team; making some graphs more explanatory; improving the layout of the data/graphics presentation; the documentation that informs which fields are required and expected by the system for the evaluator to prepare its log file; and, offer help in creating new queries. Along with those suggested improvements, we recommend carrying out usability evaluations.

In this thesis, we have applied HCI-ON in ontology-oriented software development, other applications can be explored in future works, such as the ones discussed below.

- **Harmonize Standards**: different standards often adopt different terminologies and, sometimes, it is necessary to use different standards in a combined way. That means it is necessary to harmonize different terminologies/conceptualizations so that different standards can work together and knowledge workers can better understand each other. We present in Chapter 1 problems identified in some international standards (ISO/IEC/IEEE, 2017; ISO/IEC, 2014; ISO/IEC, 2008b; ISO/IEC/IEEE, 2018; ISO/IEC, 2012; ISO/IEC, 2016; ISO, 2019b) that can use HCI-ON to harmonize such standards. For example, consider the three different definitions for "user interface" provided by (ISO/IEC, 2008b;

ISO/IEC, 2012; ISO/IEC, 2014; ISO/IEC/IEEE, 2018; ISO/IEC/IEEE, 2017) and presented in Section 1.1. As discussed, in those definitions it is not clear if interactive system, computer system and system have the same meaning. As a consequence, the definition of user interface is also not clear: does any system have user interface? By using HCIO as a reference conceptualization, it is possible to conclude that in those definitions interactive system, computer system and system have the same meaning, which is semantically equivalent to the Interactive Computer System concept in HCIO, because, according to HCIO, only interactive computer systems have user interface.

- **Communication & Knowledge**: HCI-ON can also be useful to knowledge workers for **communication** purposes. In a very simple application, the conceptualization provided by HCI-ON can serve as a reference when talking about HCI, preventing different knowledge workers (e.g., designers and users) from different understandings about the same concept in the domain and, at same time, allowing a more comprehensive understanding about it. In a more sophisticated approach, HCI-ON can be used to support **knowledge-related solutions**, such as knowledge management (KM) systems, aiding in knowledge representation (e.g., semantic annotation), integration, search, and retrieval. HCI-ON conceptual model can also be used as a basis to design KM systems integrating several HCI sub-domains. For example, HCI-ON can be used in the development of an ontology-based KM infrastructure for an HCI Environment to support activities performed in the context of an HCI project, such as HCI design and evaluation. Furthermore, it can be helpful for structuring, storing and managing knowledge items (such as lessons learned and best practices).

- **Semantic Documentation**: another possible application of HCI-ON regards **semantic documentation**. Semantic documents aim at combining documents and ontologies to provide metadata for documents and allowing users to access their knowledge in multiple ways (Eriksson, 2007). HCI-ON can be used to annotate HCI-related documents (e.g., text document, spreadsheets, images), allowing easily to retrieve and integrate information from these documents in an infrastructure for managing semantic documents as the one proposed in (Arantes; Falbo, 2010). It can make it possible, for example, to keep traceability between software requirements, HCI design components that meet the requirements and results of the evaluation of HCI components against those requirements.

- **Systems Integration**: it is possible to extract the portion of HCI-ON to be used to integrate systems and apply it by using approaches such as (Calhau; Falbo, 2010; Renault; Barcellos; Falbo, 2018), which guide about how to use ontologies to semantically integrate systems.

- **Ontology-oriented Software Development**: similar to UXON (Manso, 2022), KTID (Castro et al., 2021; Sessa, 2021) and SNOPI (Scalser, 2022), HCI-ON extracts can be used to support the development of other systems, both at dev-time and run-time, to solve problems in the HCI domain (Happel; Seedorf, 2006).

- **Ontology-based HCI Learning**: we believe the body of ontologies composing HCI-ON can be applied for HCI learning. HCI disciplines can be better understood with the support of consolidated conceptual models as ontologies are. Moreover, it has the potential to complement HCI teaching. Teachers can adopt ontologies as complementary material in their disciplines, present the graphical model and use the conceptualization provided to minimize students' difficulties in understanding the domain, use instantiations to demonstrate their real functioning, and create new instantiations to stimulate students' thinking. Moreover, the creation of other network extracted fragments into the operational version can also open and extend the line of applications for HCI-ON in a Semantic Web (such as publishing reusable HCI knowledge resources, providing services for searching, reasoning, and querying). The more HCI-ON is applied, the more feedback is used to improve it.

Finally, there are also some improvements that can be made in UXON:

- **Coverage:** as suggested by the survey findings, extending the system context to encompass UX evaluations of other interaction logging applications.

- **General Improvements:** the results of the study indicated some general improvements: data processing time; making some graphs more explanatory; improving the layout of the data/graphics presentation; provide documentation that informs which fields are required and expected by the system for the evaluator to prepare its log file; and, offer help in creating new queries.

References

Abowd, G. D.; Beale, R. Users, systems and interfaces: A unifying framework for interaction. In: Diaper, D.; Hammond, N. (Ed.). *Proceedings of the Sixth Conference of the British Computer Society Human Computer Interaction Specialist Groupe - People and Computers VI*. Heriot-Watt University, Edinburgh, UK: Cambridge University Press, 1991. p. 73–87. ISBN 0521416949. Cited on page 103.

Albert, W.; Tullis, T. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. 2th. ed. : Morgan Kaufmann, 2013. 322 p. Cited on page 40.

Amazonas, M. et al. Composing through Interaction: a framework for collaborative music composition based on human interaction on public spaces. In: *Proceedings of Brazilian Symposium on Musical Computing (SBCM 2019)*. Sociedade Brasileira de Computação - SBC, 2019. p. 42–49. Available on: <https://sol.sbc.org.br/index.php/sbcm/article/view/10421>. Cited 2 times on pages 143 e 144.

Arango-López, J. et al. Modeling and defining the pervasive games and its components from a perspective of the player experience. In: Rocha, Á. et al. (Ed.). *Proceedings of the Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, vol 746*. Cham: Springer International Publishing, 2018. p. 625–635. ISBN 978-3-319-77712-2. Cited on page 62.

Arantes, L. d. O.; Falbo, R. d. A. An Infrastructure for Managing Semantic Documents. In: *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*. Vitória. Brazil: IEEE, 2010. p. 235–244. ISBN 978-1-4244-7965-8. Available on: <http://ieeexplore.ieee.org/document/5629043/>. Cited on page 174.

Bakaev, M.; Avdeenko, T. Ontology to support web design activities in e-commerce software development process. In: *ACIT - Information and Communication Technology*. Calgary,AB,Canada: ACTAPRESS, 2010. p. 241–248. ISBN 978-0-88986-841-0. Available on: <http://www.actapress.com/PaperInfo.aspx?paperId=41151>. Cited on page 61.

Bakaev, M.; Avdeenko, T. User interface design guidelines arrangement in a recommender system with frame ontology. In: *Proceedings of the Database Systems for Advanced Applications. DASFAA 2012. Lecture Notes in Computer Science, vol 7240*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 311–322. ISBN 978-3-642-29023-7. Cited on page 61.

Bakaev, M.; Gaedke, M. Application of evolutionary algorithms in interaction design: From requirements and ontology to optimized web interface. In: *2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW)*. St. Petersburg, Russia: IEEE, 2016. p. 129–134. ISBN 978-1-5090-0445-4. Available on: <http://ieeexplore.ieee.org/document/7448138/>. Cited on page 62.

Bakaev, M. et al. Integration platform for metric-based analysis of web user interfaces. In: *Proceedings of the International Conference on Web Engineering. ICWE 2019. Lecture Notes in Computer Science, vol 11496*. Cham: Springer International Publishing, 2019. p. 525–529. ISBN 978-3-030-19274-7. Cited on page 62.

Barbosa, S. D. J.; Silva, B. S. *Interação Humano-Computador*. 1st editio. ed. Rio de Janeiro: Elsevier, 2010. 408 p. ISBN 978-85-352-3418-3. Cited 3 times on pages 33, 35 e 36.

Barbosa, S. D. J. et al. *Interação Humano-Computador e Experiência do Usuário*. : Autopublicação, 2021. Cited 3 times on pages 38, 39 e 40.

Barcellos, M.; Falbo, R. d. A.; Frauches, V. G. V. Towards a measurement ontology pattern language. *Proceedings of the 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering*, v. 1301, 2014. Cited 5 times on pages 40, 52, 126, 131 e 144.

Basili, V. R.; Caldiera, G.; Rombach, H. D. The Goal Question Metric Approach. In: . 1994. Cited on page 152.

Belkhiria, C.; Peysakhovich, V. Electro-Encephalography and Electro-Oculography in Aeronautics: A Review Over the Last Decade (2010–2020). *Frontiers in Neuroergonomics*, v. 1, p. 3, dec 2020. ISSN 2673-6195. Available on: <https://www.frontiersin.org/article/10.3389/fnrgo.2020.606719https://www.frontiersin.org/articles/10.3389/fnrgo.2020.606719/full>. Cited on page 103.

Benyon, D. *Designing interactive systems A comprehensive guide to HCI and interaction design*. 2nd. ed. : Pearson Education, 2010. 712 p. ISBN 9780321435330. Cited 3 times on pages 34, 103 e 125.

Bevan, N. International Standards for HCI. In: *Encyclopedia of Human Computer Interaction*. IGI Global, 2006. p. 362–372. Available on: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-562-7.ch056>. Cited 2 times on pages 36 e 37.

Bezold, M.; Minker, W. A framework for adapting interactive systems to user behavior. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 2, n. 4, p. 369–387, dec 2010. ISSN 18761364. Available on: <http://dl.acm.org/citation.cfm?id=2021081.2021084https://www.medra.org/servlet/aliasResolver?alias=iospress{&}doi=10.3233/AIS-2010-0>. Cited on page 61.

Bigham, J. P.; Bernstein, M. S.; Adar, E. Human-computer interaction and collective intelligence. In: Malone, T. W.; Bernstein, M. S. (Ed.). *The Collective Intelligence Handbook*. Cambridge: MIT Press, 2015. ISBN 978-0-262-02981-0. Cited on page 23.

Borgo, S.; Masolo, C. Foundational choices in dolce. In: ____. *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 361–381. ISBN 978-3-540-92673-3. Doi:10.1007/978-3-540-92673-3_16. Cited on page 41.

Borst, W. N. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Tese (Doutorado) — University of Twente, Netherlands, set. 1997. Cited on page 40.

Bowen, J.; Hinze, A. Using ontologies to reason about the usability of interactive medical devices in multiple situations of use. In: *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems - EICS '12*. New York, New York, USA: ACM Press, 2012. (EICS '12), p. 247. ISBN 9781450311687. Available on: <http://dl.acm.org/citation.cfm?doid=2305484.2305525>. Cited 2 times on pages 61 e 78.

Brank, J.; Grobelnik, M.; Mladenić, D. A survey of ontology evaluation techniques. In:

Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005). Ljubljana, Slovenia: , 2005. p. 166–170. Cited 4 times on pages 68, 70, 76 e 114.

Bringuente, A. C. d. O.; Falbo, R. d. A.; Guizzardi, G. Using a foundational ontology for reengineering a software process ontology. *Journal of Information and Data Management*, v. 2, n. 3, p. 511–526, 2011. Cited 7 times on pages 41, 48, 50, 52, 104, 125 e 126.

Calhau, R. F.; Falbo, R. d. A. An Ontology-Based Approach for Semantic Integration. In: *Proceedings of the 2010 14th IEEE International Enterprise Distributed Object Computing Conference*. Vitória. Brazil: IEEE, 2010. p. 111–120. ISSN 1541-7719. Cited on page 174.

Callan, D. E. et al. The Brain Is Faster than the Hand in Split-Second Intentions to Respond to an Impending Hazard: A Simulation of Neuroadaptive Automation to Speed Recovery to Perturbation in Flight Attitude. *Frontiers in Human Neuroscience*, v. 10, p. 187, apr 2016. ISSN 1662-5161. Available on: <https://www.frontiersin.org/article/10.3389/fnhum.2016.00187http://journal.frontiersin.org/Article/10.3389/fnhum.2016.00187/abstract>. Cited on page 103.

Carbonera, J. L.; Abel, M.; Scherer, C. M. Visual interpretation of events in petroleum exploration: An approach supported by well-founded ontologies. *Expert Systems with Applications*, v. 42, n. 5, p. 2749–2763, apr 2015. ISSN 09574174. Available on: <https://www.sciencedirect.com/science/article/pii/S0957417414007076https://linkinghub.elsevier.com/retrieve/pii/S0957417414007076>. Cited on page 24.

Carmagnola, F.; Cena, F.; Gena, C. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, Kluwer Academic Publishers, USA, v. 21, n. 3, p. 285–331, aug 2011. ISSN 0924-1868. Available on: <https://doi.org/10.1007/s11257-011-9097-5http://link.springer.com/10.1007/s11257-011-9097-5>. Cited on page 23.

Carroll, J. M. HUMAN-COMPUTER INTERACTION: Psychology as a Science of Design. *Annual Review of Psychology*, v. 48, n. 1, p. 61–83, feb 1997. ISSN 0066-4308. Available on: <https://doi.org/10.1146/annurev.psych.48.1.61http://www.annualreviews.org/doi/10.1146/annurev.psych.48.1.61>. Cited on page 33.

Carroll, J. M. Human Computer Interaction (HCI). In: Soegaard, M.; Dam, R. F. (Ed.). *The Encyclopedia of Human-Computer Interaction*. 2nd. ed. Aarhus, Denmark: The Interaction Design Foundation, 2014. Chap. 2, p. 21–61. ISBN 978-87-92964-00-7. Cited 4 times on pages 21, 33, 38 e 103.

Castro, M. V. H. B.; Barcellos, M. P.; Falbo, R. An Ontological View of Design in the Software Context. In: *14th Seminar on Ontology Research in Brazil (ONTOBRAS)*. : CEUR Workshop Proceedings, 2021. Cited on page 87.

Castro, M. V. H. B. et al. Using Ontologies to aid Knowledge Sharing in HCI Design. In: *Proceedings of the XX Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2021. p. 1–7. ISBN 9781450386173. Doi:10.1145/3472301.3484327. Available on: <https://dl.acm.org/doi/10.1145/3472301.3484327>. Cited 7 times on pages 12, 30, 32, 163, 164, 169 e 174.

Castro, M. V. H. B. et al. Knowledge Management in Human-Computer Interaction Design: A Mapping Study. In: *Proceedings of the XXIII Iberoamerican Conference on Software Engineering, CIbSE 2020, Curitiba, Brazil, November*. Curitiba, Brazil: , 2020. p. 9–13. Cited on page 23.

Castro, M. V. H. B. et al. Investigating Knowledge Management in Human-Computer Interaction Design. *Journal of Software Engineering Research and Development*, v. 9, n. 1, p. 20, 2022. Available on: <https://sol.sbc.org.br/journals/index.php/jserd/article/view/1878>. Cited on page 163.

Castro, M. V. H. B. C. *An Ontology to support Knowledge Management Solutions for Human-Computer Interaction Design*. Dissertação (Mestrado) — Computer Science Department, Federal University of Espírito Santo (UFES), 2021. Cited 6 times on pages 85, 87, 125, 163, 164 e 168.

Celino, I.; Corcoglioniti, F. Towards the formalization of interaction semantics. In: *Proceedings of the 6th International Conference on Semantic Systems - I-SEMANTICS '10*. New York, New York, USA: ACM Press, 2010. (I-SEMANTICS '10), p. 1. ISBN 9781450300148. Available on: <http://portal.acm.org/citation.cfm?doid=1839707.1839719>. Cited on page 61.

Chammas, A.; Quaresma, M.; Mont'Alvão, C. A Closer Look on the User Centred Design. *Procedia Manufacturing*, v. 3, p. 5397–5404, 2015. ISSN 2351-9789. Available on: <http://www.sciencedirect.com/science/article/pii/S2351978915006575>. Cited on page 36.

Chera, C.-M.; Tsai, W.-T.; Vatavu, R.-D. Gesture ontology for informing Service-oriented Architecture. In: *Proceedings of 2012 IEEE International Symposium on Intelligent Control*. Dubrovnik, Croatia: IEEE, 2012. p. 1184–1189. ISBN 978-1-4673-4600-9. ISSN 2158-9879. Available on: <http://ieeexplore.ieee.org/document/6398257/>. Cited on page 61.

Clemmensen, T. Hci and interoperability. In: . 2018. Null ; Conference date: 15-03-2018 Through 15-03-2018. Available on: <http://ifip-tc13.org/tc13-tc14-open-symposium-2018-tallinn-estonia/>. Cited on page 22.

Clites, T. R. et al. Proprioception from a neurally controlled lower-extremity prosthesis. *Science Translational Medicine*, American Association for the Advancement of Science, v. 10, n. 443, p. eaap8373, may 2018. ISSN 1946-6234. Available on: <https://stm.sciencemag.org/content/10/443/eaap8373https://stm.sciencemag.org/lookup/doi/10.1126/scitranslmed.aap8373>. Cited on page 103.

Coronato, A.; DE PIETRO, G. A context service for multimodal pervasive environments. In: *Proceedings of the 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*. Nice, France, France: IEEE, 2007. p. 84–87. ISBN 978-1-4244-0974-7. Available on: <http://ieeexplore.ieee.org/document/4550312/>. Cited on page 60.

Costa, C.; Murta, L. Version control in Distributed Software Development: A systematic mapping study. In: *Proceedings of the IEEE 8th International Conference on Global Software Engineering, ICGSE 2013*. 2013. p. 90–99. ISBN 9780768550572. Cited on page 58.

Costa, S. D.; Barcellos, M. P. *HCIEO dictionary of terms*. 2021. 3 p. Available at <http://bit.ly/HCIEOdic>. Cited on page 136.

Costa, S. D.; Barcellos, M. P.; Falbo, R. d. A. *HCIO complete instantiation & dictionary of terms*. 2020. 8 p. Available at <http://bit.ly/HCIO_instantiation>. Cited on page 116.

Costa, S. D.; Barcellos, M. P.; Falbo, R. d. A. *Human-Computer Interaction Cognitive Engineering Ontology (HCICEO) - specification & instantiation & dictionary of terms*. 2020. 27 p. Available at <http://bit.ly/_HCICEO> (in Portuguese only). Cited on page 120.

Costa, S. D.; Barcellos, M. P.; Falbo, R. d. A. *Ontologies in HCI - Protocol & Publications & Data Extraction*. 2020. Available at <http://bit.ly/_hciR>. Cited 4 times on pages 59, 60, 75 e 171.

Costa, S. D.; Barcellos, M. P.; Falbo, R. d. A. Ontologies in human–computer interaction: A systematic literature review. *Applied Ontology*, p. 1–32, oct 2021. ISSN 18758533. Doi:10.3233/AO-210255. Cited 6 times on pages 24, 28, 30, 56, 167 e 171.

Costa, S. D. et al. Towards an Ontology Network on Human-Computer Interaction. In: Dobbie, G. et al. (Ed.). *Proceedings of the 39th International Conference on Conceptual Modeling*. Cham: Springer International Publishing, 2020. p. 331–341. ISBN 978-3-030-62522-1. Doi:10.1007/978-3-030-62522-1_24. Cited 10 times on pages 24, 29, 49, 79, 84, 85, 125, 163, 168 e 169.

Costa, S. D. et al. A core ontology on the Human–Computer Interaction phenomenon. *Data & Knowledge Engineering*, v. 138, p. 101977, mar 2022. ISSN 0169023X. Doi:10.1016/j.datak.2021.101977. Cited 3 times on pages 29, 101 e 170.

D'Aquin, M.; Gangemi, A. Is there beauty in ontologies? *Applied Ontology*, v. 6, n. 3, p. 165–175, 2011. ISSN 15705838. Doi:10.3233/AO-2011-0093. Cited 5 times on pages 29, 71, 76, 121 e 139.

Devaurs, D.; Rath, A. S.; Lindstaedt, S. N. EXPLOITING THE USER INTERACTION CONTEXT FOR AUTOMATIC TASK DETECTION. *Applied Artificial Intelligence*, v. 26, n. 1-2, p. 58–80, jan 2012. ISSN 0883-9514. Available on: <http://www.tandfonline.com/doi/abs/10.1080/08839514.2012.629522>. Cited on page 61.

Dix, A. et al. *Human-Computer Interaction*. 3rd. ed. Upper Saddle River, NJ, USA: Pearson Prentice-Hall, Inc., 2004. 834 p. ISBN 978-0130461094. Cited 3 times on pages 34, 103 e 125.

Duarte, B. et al. Towards an Ontology of Requirements at Runtime. In: *Proceedings of the 9th International Conference on Formal Ontology in Information Systems*. Annecy, France: , 2016. Cited on page 42.

Duarte, B. B. et al. Ontological foundations for software requirements with a focus on requirements at runtime. *Applied Ontology*, IOS Press, v. 13, n. 2, p. 73–105, 2018. Cited 3 times on pages 50, 104 e 125.

Edwards, W. K.; Newman, M. W.; Poole, E. S. The infrastructure problem in HCI. In: *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, 2010. (CHI '10), p. 423. ISBN 9781605589299. Cited on page 22.

Elyusufi, Y.; Seghiouer, H.; Alimam, M. A. Building profiles based on ontology for recommendation custom interfaces. In: *2014 International Conference on Multimedia Computing and Systems (ICMCS)*. Marrakech, Morocco: IEEE, 2014. p. 558–562. ISBN 978-1-4799-3824-7. Available on: <http://ieeexplore.ieee.org/document/6911166/>. Cited on page 62.

Eriksson, H. The semantic-document approach to combining documents and ontologies. *International Journal of Human Computer Studies*, v. 65, n. 7, p. 624–639, 2007. ISSN 10715819. Cited on page 174.

ES, E. *Human Factors (HF); Personalization and User Profile Management; User Profile Preferences and Information*. 2010. Cited on page 72.

Fairclough, S. H. Fundamentals of Physiological Computing. *Interact. Comput.*, Elsevier Science Inc., USA, v. 21, n. 1–2, p. 133–145, jan 2009. ISSN 0953-5438. Available on: <https://doi.org/10.1016/j.intcom.2008.10.011>. Cited on page 103.

Fairclough, S. H.; Gilleade, K. e. *Advances in Physiological Computing*. 1. ed. : Springer-Verlag London, 2014. 254 p. (Human–Computer Interaction Series). ISBN 978-1-4471-6391-6,978-1-4471-6392-3. Cited on page 103.

Falbo, R. d. A. SABiO: Systematic approach for building ontologies. In: *Proceedings of the CEUR Workshop Proceedings*. Rio de Janeiro, Brazil: , 2014. p. 14. ISSN 16130073. Cited 7 times on pages 104, 114, 116, 121, 126, 134 e 135.

Falbo, R. d. A. et al. Organizing Ontology Design Patterns as Ontology Pattern Languages. In: Cimiano, P. et al. (Ed.). *Proceedings of the The Semantic Web: Semantics and Big Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 61–75. ISBN 978-3-642-38288-8. Cited 2 times on pages 11 e 41.

Feilmayr, C.; Wöß, W. An analysis of ontologies and their success factors for application to business. *Data & Knowledge Engineering*, v. 101, p. 1–23, 2016. ISSN 0169-023X. Cited on page 42.

Fernández-López, M.; Gómez-Pérez, A.; Juristo, N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*. American Asociation for Artificial Intelligence, 1997. Available on: <http://oa.upm.es/5484/>. Cited 2 times on pages 68 e 140.

Gangemi, A. et al. Sweetening Ontologies with DOLCE. In: Gómez-Pérez, A.; Benjamins, V. R. (Ed.). *Proceedings of the Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 166–181. ISBN 978-3-540-45810-4. Cited 2 times on pages 41 e 72.

Garay-Vitoria, N.; Cearreta, I.; Larraza-Mendiluze, E. Application of an Ontology-Based Platform for Developing Affective Interaction Systems. *IEEE Access*, v. 7, p. 40503–40515, 2019. ISSN 2169-3536. Available on: <https://ieeexplore.ieee.org/document/8662572/>. Cited on page 62.

Gómez-Pérez, A.; Suárez-Figueroa, M. C. NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology. In: *Proceedings of the International Conference on Software, Services & Semantic technologies (S3T 2009)*. Bulgaria: Universidad de Sofia, 2009. Available on: <http://oa.upm.es/5475/>. Cited 2 times on pages 68 e 140.

Grenon, P.; Smith, B.; Goldberg, L. Biodynamic Ontology: Applying BFO in the Biomedical Domain. *Studies in health technology and informatics*, v. 102, p. 20–38, 2004. Cited on page 72.

Gruber, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, Academic Press Ltd., GBR, v. 5, n. 2, p. 199–220, jun 1993. ISSN 10428143. Cited on page 40.

Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, Academic Press, Inc., Duluth, MN, USA, v. 43, n. 5-6, p. 907–928, nov 1995. ISSN 10715819. Available on:

<https://linkinghub.elsevier.com/retrieve/pii/S1071581985710816>. Cited 2 times on pages 24 e 40.

Guarino, N. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. 1st. ed. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1998. ISBN 9051993994. Cited 3 times on pages 28, 40 e 41.

Guarino, N.; Oberle, D.; Staab, S. What Is an Ontology? In: *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1–17. ISBN 978-3-540-92673-3. Cited on page 40.

Guizzardi, G. *Ontological foundations for structural conceptual models*. Tese (Doutorado) — University of Twente, 2005. Cited 10 times on pages 28, 29, 31, 33, 42, 43, 81, 83, 104 e 125.

Guizzardi, G. On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In: Vasilecas, O.; Edler, J.; Caplinskas, A. (Ed.). *Proceedings of the Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 18–39. ISBN 978-1-58603-640-8. Available on: <http://dl.acm.org/citation.cfm?id=1565421.1565425>. Cited 3 times on pages 28, 42 e 122.

Guizzardi, G.; Falbo, R.; Guizzardi, R. S. S. Grounding software domain ontologies in the unified foundational ontology (ufo): The case of the ode software process ontology. In: *Proceedings of the 1th Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'2008)*. 2008. p. 127–140. Cited 7 times on pages 28, 29, 31, 42, 43, 48 e 81.

Guizzardi, G. et al. Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0. In: Trujillo, J. C. et al. (Ed.). *Proceedings of the Conceptual Modeling*. Xi'an, China: Springer, Cham, 2018. p. 136–150. ISBN 978-3-030-00847-5. Cited 5 times on pages 28, 29, 31, 42 e 43.

Guizzardi, G. et al. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Applied Ontology*, 2015. ISSN 18758533. Cited 4 times on pages 28, 29, 31 e 43.

Guizzardi, G. et al. Towards Ontological Foundations for the Conceptual Modeling of Events. In: Ng, W.; Storey, V. C.; Trujillo, J. C. (Ed.). *Proceedings of the Conceptual Modeling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 327–341. ISBN 978-3-642-41924-9. Cited 6 times on pages 28, 29, 31, 42, 43 e 81.

Haller, A.; Groza, T.; Rosenberg, F. Interacting with linked data via semantically annotated widgets. In: *Proceedings of the The Semantic Web, . JIST 2011. Lecture Notes in Computer Science, vol 7185*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 300–317. ISBN 978-3-642-29923-0. Cited on page 61.

Happel, H.-J.; Seedorf, S. Applications of Ontologies in Software Engineering. *Proceedings of the 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006), held at the 5th International Semantic Web Conference (ISWC 2006)*, p. 14, 2006. Cited 2 times on pages 146 e 174.

Harper, R. et al. *Being Human - Human-Computer Interaction in the Year 2020*. 2008. ISSN 1527-6619. ISBN 978-0-9554761-1-2. Cited on page 21.

Harrison, S.; Tatar, D.; Sengers, P. The three paradigms of HCI. *Alt. Chi. Session at the SIGCHI ...*, 2007. ISSN 07374038. Cited on page 34.

Hassan, H. M.; Galal-Edeen, G. H. From usability to user experience. In: *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*. IEEE, 2017. p. 216–222. ISBN 978-1-5090-6664-3. Available on: <http://ieeexplore.ieee.org/document/8279761/>. Cited 2 times on pages 38 e 39.

Hassenzahl, M.; Tractinsky, N. User experience - a research agenda. *Behaviour & Information Technology*, Taylor & Francis, v. 25, n. 2, p. 91–97, 2006. Available on: <https://doi.org/10.1080/01449290500330331>. Cited on page 36.

Hevner, A. R. A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, v. 19, n. 2, p. 87–92, 2007. Available on: <https://aisel.aisnet.org/sjis/vol19/iss2/4>. Cited 5 times on pages 11, 26, 27, 28 e 29.

Hevner, A. R. et al. Design Science in Information Systems Research. *MIS Q.*, Society for Information Management and The Management Information Systems Research Center, USA, v. 28, n. 1, p. 75–105, mar 2004. ISSN 0276-7783. Cited 3 times on pages 11, 26 e 27.

Hewett, T. et al. *ACM SIGCHI Curricula for Human-Computer Interaction*. New York, NY, USA: Association for Computing Machinery, 1992. 173 p. ISBN 0897914740. Available on: <http://dl.acm.org/citation.cfm?id=2594128>. Cited 2 times on pages 33 e 103.

Hornbæk, K.; Oulasvirta, A. What Is Interaction? In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017. p. 5040–5052. ISBN 9781450346559. Doi:10.1145/3025453.3025765. Cited on page 34.

Ibrahim, R.; Fay, R. Enhancing cognition by understanding knowledge flow characteristics during design collaboration. *ALAM CIPTA, Intl. J. on Sustainable Tropical Design Research & Practice*, v. 1, n. 1, p. 9–16, 2006. Cited on page 23.

ISO. *Assistive products for persons with disability — Classification and terminology*. 2011. 100 p. Available on: <https://www.iso.org/standard/50982.html>. Cited on page 72.

ISO. *Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. 2018. 36 p. Cited on page 125.

ISO. *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. 2019. 33 p. Available on: <https://www.iso.org/standard/77520.html>. Cited 6 times on pages 11, 36, 37, 38, 103 e 125.

ISO. *Ergonomics of human-system interaction — Part 220: Processes for enabling, executing and assessing human-centred design within organizations*. 2019. 90 p. Available on: <https://www.iso.org/standard/63462.html>. Cited 3 times on pages 22, 125 e 173.

ISO/IEC. *Information technology — Software product evaluation — Part 5: Process for evaluators*. 1998. 35 p. Available on: <https://www.iso.org/standard/24906.html>. Cited on page 125.

ISO/IEC. *Software engineering — Product quality — Part 1: Quality model*. 2001. 25 p. Available on: <https://www.iso.org/standard/22749.html>. Cited on page 125.

ISO/IEC. *Information technology — Individualized adaptability and accessibility in e-learning, education and training — Part 1: Framework and reference model*. 2008. 34 p. Available on: <https://www.iso.org/standard/41521.html>. Cited on page 72.

ISO/IEC. *Systems and software engineering–requirements for designers and developers of user documentation*. 2008. 143 p. Available on: <https://www.iso.org/standard/43073.html>. Cited 3 times on pages 22, 173 e 174.

ISO/IEC. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*. 2011. 45 p. Available on: <https://www.iso.org/standard/35765.html>. Cited on page 125.

ISO/IEC. *Information technology – Object Management Group Architecture-Driven Modernization (ADM) – Knowledge Discovery Meta-Model (KDM)*. 2012. 331 p. Available on: <https://www.iso.org/standard/32625.html>. Cited 3 times on pages 22, 173 e 174.

ISO/IEC. *Systems and software engineering–Systems and software product Quality Requirements and Evaluation (SQuaRE) Common Industry Format (CIF) for usability: Context of use description*. 2014. 33 p. Available on: <https://www.iso.org/standard/35789.html>. Cited 3 times on pages 22, 173 e 174.

ISO/IEC. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for Usability — Evaluation Report*. 2016. 37 p. Available on: <https://www.iso.org/standard/63831.html>. Cited 3 times on pages 22, 125 e 173.

ISO/IEC TR. *Systems and software engineering–Systems and software product Quality Requirements and Evaluation (SQuaRE)–Common Industry Format (CIF) for usability: General framework for usability-related information*. 2010. 22 p. Available on: <https://www.iso.org/standard/35786.html>. Cited on page 125.

ISO/IEC/IEEE. *International Standard - Systems and software engineering–Vocabulary*. 2017. 1–541 p. Available on: <https://pascal.computer.org/https://ieeexplore.ieee.org/document/8016712>. Cited 3 times on pages 22, 173 e 174.

ISO/IEC/IEEE. *Systems and software engineering–Requirements for acquirers and suppliers of information for users*. 2018. 37 p. Available on: <https://www.iso.org/standard/72088.html>. Cited 3 times on pages 22, 173 e 174.

ISO/TR. *Ergonomics of human-system interaction — Usability methods supporting human-centred design*. 2002. 44 p. Available on: <https://www.iso.org/standard/31176.html>. Cited on page 125.

ISO/TS. *Ergonomics of human-system interaction — Specification for the process assessment of human-system issues* . 2010. 92 p. Available on: <https://www.iso.org/standard/56174.html>. Cited on page 125.

Kabir, M. A. et al. Inferring User Situations from Interaction Events in Social Media. *The Computer Journal*, v. 58, n. 9, p. 2026–2043, sep 2015. ISSN 0010-4620. Available on: <http://doi.org/10.1093/comjnl/bxu131>. Cited on page 62.

Karim, S.; Tjoa, A. M. Towards the Use of Ontologies for Improving User Interaction for People with Special Needs. In: Miesenberger, K. et al. (Ed.). *Proceedings of the International*

Conference on Computers for Handicapped Persons (ICCHP) Computers Helping People with Special Needs. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 77–84. ISBN 978-3-540-36021-6. Cited on page 60.

Kendall, K. E.; Kendall, J. E. Human-computer interaction. In: *Systems Analysis and Design*. 7th. ed. Upper Saddle River, NJ, USA: Pearson College Div, 2008. Chap. 14, p. 774. ISBN 978-0132240857. Cited on page 23.

Kitchenham, B. A.; Budgen, D.; Brereton, O. P. Using Mapping Studies As the Basis for Further Research - A Participant-observer Case Study. *Information and Software Technology*, Butterworth-Heinemann, Newton, MA, USA, v. 53, n. 6, p. 638–651, jun 2011. ISSN 0950-5849. Available on: <http://dx.doi.org/10.1016/j.infsof.2010.12.011>. Cited on page 56.

Kitchenham, B. A.; Charters, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007. v. 2.3, 65 p. Cited 3 times on pages 56, 57 e 77.

Klyne, G. et al. *Composite capability/preference profiles (cc/pp): structure and vocabularies 1.0*. 2004. Available on: <https://www.w3.org/TR/CCPP-struct-vocab/>. Cited on page 72.

Korfiatis, N.; Constantiou, I. Socially targeted mobile services: Towards an upper level ontology of social roles for mobile environments. In: *Proceedings of the 14th European Conference on Information Systems*. Göteborg; Sweden: , 2006. p. 12. Available on: <https://aisel.aisnet.org/ecis2006/60>. Cited on page 60.

Koutkias, V. et al. An integrated semantic framework supporting universal accessibility to ICT. *Universal Access in the Information Society*, Springer-Verlag, Berlin, Heidelberg, v. 15, n. 1, p. 49–62, mar 2016. ISSN 1615-5289. Available on: <http://dx.doi.org/10.1007/s10209-014-0372-1http://link.springer.com/10.1007/s10209-014-0372-1>. Cited on page 62.

Krol, L. R. et al. A task-independent workload classifier for neuroadaptive technology: Preliminary data. In: *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2016. p. 3171–3174. Cited on page 103.

Kultsova, M. et al. A Two-Phase Method of User Interface Adaptation for People with Special Needs. In: Kravets, A. et al. (Ed.). *Proceedings of the Conference on Creativity in Intelligent Technologies and Data Science. CIT&DS 2017. Communications in Computer and Information Science, vol 754.* Volgograd; Russia: Springer, Cham, 2017. p. 805–821. ISBN 978-3-319-65551-2. Cited on page 62.

Landis, J. R.; Koch, G. G. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, [Wiley, International Biometric Society], v. 33, n. 1, p. 159–174, sep 1977. ISSN 0006341X, 15410420. Available on: <http://www.jstor.org/stable/2529310>. Cited 2 times on pages 75 e 76.

Lebib, F.-Z.; Mellah, H.; Mohand-Oussaid, L. Ontological Interaction Modeling and Semantic Rule-based Reasoning for User Interface Adaptation. In: INSTICC. *Proceedings of the 12th International Conference on Web Information Systems and Technologies*. Rome, Italy: SCITEPRESS - Science and and Technology Publications, 2016. p. 347–354. ISBN 978-989-758-186-1. Available on: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005854303470354>. Cited on page 62.

Liyanage, H.; Krause, P.; Lusignan, S. de. Using ontologies to improve semantic interoperability in health data. *BMJ Health & Care Informatics*, BMJ Specialist Journals, v. 22, n. 2, p. 309–315, 2015. Available on: <https://informatics.bmj.com/content/22/2/309>. Cited on page 24.

Mahieu, C. et al. Semantics-based platform for context-aware and personalized robot interaction in the internet of robotic things. *Journal of Systems and Software*, v. 149, p. 138–157, mar 2019. ISSN 01641212. Available on: <http://www.sciencedirect.com/science/article/pii/S0164121218302553https://linkinghub.elsevier.com/retrieve/pii/S0164121218302553>. Cited on page 63.

Manso, C. d. F. *UXON: Um Sistema Baseado em Ontologias para Avaliação da Experiência de Usuário em Sistemas Imersivos*. 2022. In Portuguese only. Cited 9 times on pages 30, 32, 142, 144, 147, 148, 168, 171 e 174.

Mariño, B. D. R. et al. Accessibility and Activity-Centered Design for ICT Users: ACCESIBILITIC Ontology. *IEEE Access*, v. 6, p. 60655–60665, 2018. ISSN 2169-3536. Available on: <https://ieeexplore.ieee.org/document/8490866/>. Cited on page 62.

Marques, L. et al. UX-Trek: A post-interaction journey from immersive experience logs. In: *Proceedings of the XIX Brazilian Symposium on Human Factors in Computing Systems (IHC'20)*. Diamantina, Brazil: ACM, 2020. p. 6. Doi:10.1145/3424953.3426547. Cited 5 times on pages 39, 40, 143, 144 e 145.

Martín, A. et al. Engineering Accessible Web Applications. An Aspect-Oriented Approach. *World Wide Web*, v. 13, n. 4, p. 419–440, dec 2010. ISSN 1386-145X. Available on: <http://link.springer.com/10.1007/s11280-010-0091-3>. Cited 2 times on pages 61 e 78.

Menezes, T. C.; Nonnecke, B. UX-Log: Understanding Website Usability through Recreating Users' Experiences in Logfiles. *International Journal of Virtual Worlds and Human Computer Interaction*, v. 2, p. 47–56, 2014. ISSN 23686103. Available on: <http://vwhci.avestia.com/2014/006.html>. Cited on page 40.

Meng, Z. Architecture Support for Context-aware Adaptation of Rich Sensing Smartphone Applications. *KSII Transactions on Internet and Information Systems*, v. 12, n. 1, p. 248–268, jan 2018. ISSN 19767277. Available on: <http://itiis.org/digital-library/manuscript/1902>. Cited on page 62.

Mezhoudi, N.; Vanderdonckt, J. A user's feedback ontology for context-aware interaction. In: *Proceedings of the 2015 2nd World Symposium on Web Applications and Networking (WSWAN)*. Sousse, Tunisia: IEEE, 2015. p. 1–7. ISBN 978-1-4799-8172-4. Available on: <http://ieeexplore.ieee.org/document/7210331/>. Cited on page 62.

Moran, k. *Usability Testing 101*. 2019. <https://www.nngroup.com/articles/usability-testing-101/>. Cited on page 40.

Myrgioti, E.; Bassiliades, N.; Miliou, A. Bridging the HASM: An OWL ontology for modeling the information pathways in haptic interfaces software. *Expert Systems with Applications*, v. 40, n. 4, p. 1358–1371, mar 2013. ISSN 09574174. Available on: <https://linkinghub.elsevier.com/retrieve/pii/S0957417412010147>. Cited on page 61.

Nardi, J. C.; Falbo, R. d. A.; Almeida, J. P. A. A Panorama of the Semantic EAI Initiatives and the Adoption of Ontologies by these Initiatives. In: Sinderen, M. van et al. (Ed.). *Proceedings of the Enterprise Interoperability*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 198–211. ISBN 978-3-642-36796-0. Cited on page 41.

Negru, S.; Buraga, S. A Knowledge-Based Approach to the User-Centered Design Process. In: Fred, A. et al. (Ed.). *Proceedings of the Knowledge Discovery, Knowledge Engineering and Knowledge Management. IC3K 2012. Communications in Computer and Information Science, vol 415*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 165–178. ISBN 978-3-642-54105-6. Cited on page 61.

Negru, S.; Buraga, S. C. Persona Modeling Process - From Microdata-based Templates to Specific Web Ontologies. In: Filipe, J.; Dietz, J. (Ed.). *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*. Barcelona, Spain: SciTePress - Science and and Technology Publications, 2012. p. 34–42. ISBN 978-989-8565-30-3. Cited on page 61.

Nielsen, J. *Usability Engineering*. 1st. ed. San Francisco, USA: Morgan Kaufmann, 1993. 362 p. ISBN 0125184069. Cited 6 times on pages 36, 103, 119, 123, 124 e 125.

Nielsen, J. *Usability Inspection Methods*. 1st. ed. San Francisco, USA: John Wiley & Sons, 1994. 448 p. ISBN 0125184069. Cited on page 39.

Nielsen, J. *Usability Inspection Methods*. 1st. ed. San Francisco, USA: John Wiley & Sons, 1994. 448 p. ISBN 0125184069. Cited 2 times on pages 124 e 125.

Nielsen, Jakob. *10 Usability Heuristics for User Interface Design*. 2020. <https://www.nngroup.com/articles/ten-usability-heuristics/>. Cited on page 39.

Norman, D. A. Cognitive engineering. In: Norman, D. A.; Draper, S. W. (Ed.). *User Centered System Design: New Perspectives on Human-Computer Interaction*. 1. ed. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1986. Chap. 3, p. 31–61. ISBN 0898597811. Cited 2 times on pages 60 e 103.

Norman, D. A. *The Design of Everyday Things*. Reprint. New York, NY, USA: Basic Books, Inc., 2002. ISBN 9780465067107. Cited on page 36.

Norman, D. A. *The Design of Future Things*. Reprint. New York, NY, USA: Basic Books, 2009. 231 p. ISBN 9780465013036. Cited on page 103.

Norman, D. A. *The Design of Everyday Things: Revised and Expanded Edition*. 2nd. ed. Basic Books, 2013. 347 p. ISBN 9780465072996. Available on: <https://books.google.com.br/books?id=nVQPAAAAQBAJ>. Cited 2 times on pages 34 e 103.

O'Leary, D. E. Enterprise Knowledge Management. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 31, n. 3, p. 54–61, mar 1998. ISSN 0018-9162. Available on: <https://doi.org/10.1109/2.660190>. Cited on page 23.

Oliveira, R. et al. State of the Art on Formal Methods for Interactive Systems. In: Weyers, B. et al. (Ed.). *The Handbook of Formal Methods in Human-Computer Interaction*. Cham: Springer International Publishing, 2017. p. 3–55. ISBN 978-3-319-51838-1. Cited on page 103.

Pardo, C. et al. From chaos to the systematic harmonization of multiple reference models: A harmonization framework applied in two case studies. *Journal of Systems and Software*, v. 86, n. 1, p. 125–143, jan 2013. ISSN 01641212. Available on: <https://linkinghub.elsevier.com/retrieve/pii/S0164121212002282>. Cited on page 24.

Petersen, K. et al. Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swindon, GBR: BCS Learning & Development Ltd., 2008. (EASE'08), p. 68–77. Cited on page 56.

Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, Butterworth-Heinemann, Newton, MA, USA, v. 64, p. 1–18, 2015. ISSN 0950-5849. Available on: <http://www.sciencedirect.com/science/article/pii/S0950584915000646>. Cited 2 times on pages 56 e 58.

Petrie, H.; Bevan, N. The Evaluation of Accessibility, Usability, and User Experience. *The universal access handbook*, v. 1, p. 1–16, jun 2009. Available on: <http://www.crcnetbase.com/doi/abs/10.1201/9781420064995-c20>. Cited 2 times on pages 39 e 40.

Pokraev, S. V. *Model-driven semantic integration of service-oriented applications*. Tese (Doutorado) — University of Twente, oct 2009. Cited on page 21.

Pourzolfaghar, Z.; Helfert, M. Investigating HCI Challenges for Designing Smart Environments. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 9752, p. 79–90, 2016. Cited 2 times on pages 22 e 23.

Poveda-Villalón, M.; Suárez-Figueroa, M. C.; Gómez-Pérez, A. Reusing Ontology Design Patterns in a Context Ontology Network. In: *Proceedings of the 2nd International Conference on Ontology Patterns - Volume 671*. Aachen, DEU: CEUR-WS.org, 2010. (WOP'10), p. 35–52. Cited on page 79.

Preece, J.; Sharp, H.; Rogers, Y. *Interaction Design: Beyond Human-Computer Interaction*. 4th. ed. : John Wiley & Sons, 2015. 584 p. ISBN 978-1-119-02075-2. Cited 4 times on pages 21, 33, 74 e 103.

Preece, J.; Sharp, H.; Rogers, Y. *Interaction Design: Beyond Human-Computer Interaction*. 5th. ed. : John Wiley & Sons, 2019. 656 p. ISBN 978-1119547259. Cited 3 times on pages 38, 39 e 40.

Renault, L. D. C.; Barcellos, M. P.; Falbo, R. d. A. Using an Ontology-Based Approach for Integrating Applications to Support Software Processes. In: *Proceedings of the 17th Brazilian Symposium on Software Quality*. New York, NY, USA: Association for Computing Machinery, 2018. (SBQS), p. 220–229. ISBN 9781450365659. Available on: <https://doi.org/10.1145/3275245.3275269>. Cited on page 174.

Robal, T.; Marenkov, J.; Kalja, A. Ontology Design for Automatic Evaluation of Web User Interface Usability. In: *2017 Portland International Conference on Management of Engineering and Technology (PICMET)*. Portland, OR, USA: IEEE, 2017. p. 1–8. ISBN 978-1-890843-36-6. Available on: <http://ieeexplore.ieee.org/document/8125425/>. Cited on page 62.

Robson, C.; McCartan, K. *Real World Research*. 4th. ed. : Wiley, 2016. 560 p. ISBN 978-1-118-74523-6. Cited on page 159.

Rogers, Y.; Sharp, H.; Preece, J. *Interaction Design: Beyond Human-Computer Interaction*. 3rd. ed. Chichester, United Kingdom: John Wiley & Sons, 2011. 585 p. (Interaction Design: Beyond Human-computer Interaction). ISBN 9780470665763. Available on: <https://books.google.com.br/books?id=b-v{_}6BeCw>. Cited 2 times on pages 103 e 125.

Rubin, J.; Chisnell, D.; Spool, J. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. 2nd. ed. USA: John Wiley & Sons, Inc., 2008. 384 p. ISBN 0470185481. Cited 2 times on pages 124 e 125.

Runeson, P. et al. *Case study research in software engineering: Guidelines and examples*. 1st. ed. : John Wiley & Sons, 2012. 237 p. Cited on page 158.

Rus, I.; Lindvall, M. Knowledge management in software engineering. *IEEE Software*, v. 19, n. 3, p. 26–38, may 2002. ISSN 1937-4194. Cited on page 23.

Rusu, C. et al. Usability and User Experience: What Should We Care About? *International Journal of Information Technologies and Systems Approach*, v. 8, n. 2, p. 1–12, jul 2015. ISSN 1935-570X. Available on: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJITSA.2015070101>. Cited on page 21.

Ruy, F. B. *Software Engineering Standards Harmonization: An Ontology-based Approach*. 228 p. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2017. Cited 9 times on pages 11, 41, 48, 81, 89, 92, 95, 167 e 170.

Ruy, F. B. et al. SEON: A Software Engineering Ontology Network. In: Blomqvist, E. et al. (Ed.). *Proceedings of the Knowledge Engineering and Knowledge Management*. Cham: Springer International Publishing, 2016. p. 527–542. ISBN 978-3-319-49004-5. Cited 17 times on pages 11, 24, 29, 31, 33, 42, 48, 49, 50, 53, 80, 81, 87, 126, 167, 169 e 170.

Saffer, D. *Designing for Interaction: Creating Innovative Applications and Devices*. 2nd. ed. : New Riders, 2010. 240 p. ISBN 0321643399. Cited 2 times on pages 103 e 125.

Salamon, J. et al. Using Goal Modeling and OntoUML for reengineering the Good Relations Ontology. In: *Proceedings of the Brazilian Seminar on Ontologies (ONTOBRAS)*. 2017. p. 12. Cited on page 48.

Salamon, J. S. *Uma abordagem orientada a objetivos para desenvolvimento de ontologias baseado em integração*. 152 p. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2018. Available on: <http://repositorio.ufes.br/handle/10/9861>. Cited 4 times on pages 13, 89, 92 e 93.

Salma, G.; Eddine, M. E. K. K.; Sabin, M. C. B. Use of ontologies in modeling persona. In: *Proceedings of the 2012 IEEE International Conference on Complex Systems (ICCS)*. Agadir, Morocco: IEEE, 2012. p. 1–9. ISBN 978-1-4673-4766-2. Available on: <http://ieeexplore.ieee.org/document/6458521/>. Cited on page 61.

Salma, G.; Marouane, E. L. Operating an application for modeling persona by using ontologies. *Journal of Theoretical & Applied Information Technology*, v. 88, n. 1, p. 57–64, 2016. Available on: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84973549972{&}partnerID=40{&}md5=822eb166a69c5fe36c68dd8e66>. Cited on page 61.

Scalser, M. B. *SNOPI: Um Sistema de Interface Adaptativa baseada em Ontologia*. 2022. In Portuguese only. Cited 7 times on pages 12, 30, 32, 164, 165, 169 e 174.

Scherp, A. et al. Designing core ontologies. *Applied Ontology*, v. 6, n. 3, p. 177–221, 2011. ISSN 15705838. Available on: <https://www.medra.org/servlet/aliasResolver?alias=iospress{&}doi=10.3233/AO-2011-0>. Cited 3 times on pages 40, 41 e 122.

Schmidt, A. Implicit human computer interaction through context. *Personal Technologies*, v. 4, n. 2-3, p. 191–199, jun 2000. ISSN 0949-2054. Doi:10.1007/BF01324126. Cited on page 35.

Schmidt, A. Interactive Context-Aware Systems Interacting with Ambient Intelligence. *Ambient intelligence*, Citeseer, p. 159–178, 2005. Cited on page 35.

Schneider, K. *Experience and Knowledge Management in Software Engineering*. 1st. ed. Heidelberg, Berlin: Springer Publishing Company, Incorporated, 2009. 235 p. ISBN 3540958797, 9783540958796. Cited on page 23.

Sene, A.; Kamsu-Foguem, B.; Rumeau, P. Data mining for decision support with uncertainty on the airplane. *Data & Knowledge Engineering*, v. 117, p. 18–36, sep 2018. ISSN 0169023X. Available on: <https://linkinghub.elsevier.com/retrieve/pii/S0169023X1730071X>. Cited on page 24.

Sessa, B. O. *Uma Ferramenta Baseada em Ontologia para Apoiar Aspectos de Gerência de Conhecimento no Design de IHC*. 2021. In Portuguese only. Cited 4 times on pages 32, 163, 169 e 174.

Shahzad, S.; Granitzer, M.; Helic, D. Ontological model driven GUI development: User interface ontology approach. In: *Proceedings of the 6th International Conference on Computer Sciences and Convergence Information Technology, ICCIT 2011*. Seogwipo, South Korea: IEEE, 2011. p. 214–218. ISBN 9788988678541. Cited on page 61.

Silva, T. R.; Winckler, M.; Trætteberg, H. Ensuring the consistency between user requirements and gui prototypes: A behavior-based automated approach. In: *Proceedings of the Human-Computer Interaction – INTERACT 2019. INTERACT 2019. Lecture Notes in Computer Science, vol 11746*. Cham: Springer International Publishing, 2019. p. 644–665. ISBN 978-3-030-29381-9. Cited 2 times on pages 63 e 78.

Sokolowski, J.; Walczak, K. Semantic modelling of user interactions in virtual reality environments. In: *Proceedings of the Technological Innovation for Resilient Systems. DoCEIS 2018. IFIP Advances in Information and Communication Technology, vol 521*. Cham: Springer International Publishing, 2018. p. 18–27. ISBN 978-3-319-78574-5. Cited 2 times on pages 62 e 78.

Souza, C. S. *The Semiotic Engineering of Human-Computer (Acting with Technology)*. Cambridge, Massachussetts: The MIT Press, 2005. 283 p. ISSN 1098-6596. ISBN 0262042207. Cited on page 103.

Souza, É. F. d.; Falbo, R. d. A.; Vijaykumar, N. L. Knowledge management initiatives in software testing: A mapping study. *Information and Software Technology*, v. 57, p. 378–391, 2015. ISSN 0950-5849. Available on: <http://www.sciencedirect.com/science/article/pii/S0950584914001335>. Cited on page 74.

Staab, S.; Stuckenschmidt, H. (Ed.). *Semantic Web and Peer-to-Peer Decentralized Management and Exchange of Knowledge and Information*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. 365 p. ISBN 978-3-540-28346-1. Available on: <http://link.springer.com/10.1007/3-540-28347-1>. Cited on page 23.

Storrs, G. A conceptualisation of multiparty interaction. *Interacting with Computers*, v. 6, n. 2, p. 173–189, jun 1994. ISSN 09535438. Available on: <http://www.sciencedirect.com/science/article/pii/095354389490023Xhttps://academic.oup.com/iwc/article-lookup/doi/10.1016/0953-5438(94)90023-X>. Cited 2 times on pages 60 e 120.

Studer, R.; Benjamins, V.; Fensel, D. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, v. 25, n. 1-2, p. 161–197, mar 1998. ISSN 0169023X. Available on: <https://linkinghub.elsevier.com/retrieve/pii/S0169023X97000566>. Cited on page 42.

Suárez-Figueroa, M. C. et al. *Ontology Engineering in a Networked World*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. XII, 444 p. ISSN 0192-8651. ISBN 978-3-642-24793-4. Available on: <http://link.springer.com/10.1007/978-3-642-24794-1>. Cited 7 times on pages 24, 48, 49, 68, 80, 89 e 169.

Suàrez, P. R.; Jùnior, B. L.; Barros, M. A. de. Applying knowledge management in UI design process. In: Slavik, P.; Palanque, P. (Ed.). *Proceedings of the 3rd annual conference on Task models and diagrams - TAMODIA '04*. New York, New York, USA: ACM Press, 2004. p. 113. ISBN 1595930000. Available on: <http://portal.acm.org/citation.cfm?doid=1045446.1045468>. Cited 2 times on pages 60 e 78.

Sutcliffe, A. G. Requirements Engineering from an HCI Perspective. In: Soegaard, M.; Dam, R. F. (Ed.). *The Encyclopedia of Human-Computer Interaction*. 2. ed. Aarhus, Denmark: The Interaction Design Foundation, 2014. Chap. 13, p. 707–760. ISBN 978-87-92964-00-7. Cited 5 times on pages 21, 33, 35, 36 e 103.

Tamma, V. et al. Ontologies for supporting negotiation in e-commerce. *Engineering Applications of Artificial Intelligence*, v. 18, n. 2, p. 223–236, mar 2005. ISSN 09521976. Available on: <https://www.sciencedirect.com/science/article/pii/S0952197604001836https://linkinghub.elsevier.com/retrieve/pii/S0952197604001836>. Cited on page 24.

Tourwé, T. et al. Ontology-driven elicitation of multimodal user interface design recommendations. In: De Causmaecker, P. et al. (Ed.). *Proceedings of the BNAIC : Belgian/Netherlands Artificial Intelligence Conference*. Ghent, Belgium: , 2011. p. 8. ISSN 1568-7805. Available on: <http://hdl.handle.net/1854/LU-2012408>. Cited 2 times on pages 61 e 78.

Victoria, I.; Antoanela, N.; Cicortas, A. Deriving ontologies using multi-agent systems. *WSEAS Transactions on Computers*, v. 7, 2008. Cited on page 67.

Wohlin, C. et al. Systematic Literature Reviews. In: ____. *Experimentation in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 45–54. ISBN 978-3-642-29044-2. Cited on page 75.

Yago, H. et al. ON-SMMILE: Ontology Network-based Student Model for MultIple Learning Environments. *Data & Knowledge Engineering*, v. 115, p. 48–67, may 2018. ISSN 0169023X. Available on: <https://linkinghub.elsevier.com/retrieve/pii/S0169023X17301945>. Cited on page 24.

Zamzami, E.; Budiardjo, E.; Suhartanto, H. Requirements Recovery Using Ontology Model for Capturing End-to-End Interaction of Proven Application Software. *International Journal of Software Engineering and Its Applications*, v. 7, n. 6, p. 425–434, nov 2013. ISSN 17389984. Available on: <http://www.sersc.org/journals/IJSEIA/vol7{_}no6{_}2013/>. Cited on page 61.

Zander, T. O.; Kothe, C. Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of Neural Engineering*, {IOP} Publishing, v. 8, n. 2, p. 025005, apr 2011. ISSN 1741-2560. Available on: <https://doi.org/10.1088/1741-2560/8/2/025005https://iopscience.iop.org/article/10.1088/1741-2560/8/2/025005>. Cited on page 103.

Zander, T. O. et al. Enhancing Human-Computer Interaction with Input from Active and Passive

Brain-Computer Interfaces. In: ____. *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction*. London: Springer London, 2010. Chap. 11, p. 181–199. ISBN 978-1-84996-272-8. Available on: <https://doi.org/10.1007/978-1-84996-272-8{_}11http://link.springer.com/10.1007/978-1-84996-272-8>. Cited on page 103.

Zander, T. O.; Krol, L. R.; Gramann, K. Chapter 85 - Towards Neuroadaptive Technology: Implicitly Controlling a Cursor Though a Passive Brain–Computer Interface. In: Ayaz, H.; Dehais, F. (Ed.). *Neuroergonomics*. Academic Press, 2018. p. 301–302. ISBN 978-0-12-811926-6. Available on: <http://www.sciencedirect.com/science/article/pii/B9780128119266000853>. Cited on page 103.

Appendix

APPENDIX A – Artifacts used in UXON's Evaluation with User

A.1 Consent Form

Evaluation Study

Evaluation study of the use of the UXON (an ontology-based system) tool to support UX evaluation by using interaction logging information.

sidornellas@gmail.com Switch account

* Required

Email *

sidornellas@gmail.com

Consent Form

This study is being carried out in the context of the Phd's research of student Simone Dornelas Costa, carried out in the Postgraduate Program in Computer Science at the Federal University of Espirito Santo (UFES), advised by the professors Monalessa Perini Barcellos and Ricardo de Almeida Falbo (in memoriam).

In the context of my doctorate research, the study goal is to verify whether (an extract of) the Human-Computer Interaction Ontology Network (HCI-ON) can be used to produce a suitable solution for an HCI-related problem - particularly in the context of UX evaluation by using interaction logging data.

After accepting this consent form, you will be asked to answer some questions. The confidentiality of individual data provided in the study is guaranteed. The data will be used in the research context and will be not used as a personal or professional assessment. Participants' name (optional) and email are requested only for contact purposes, in case there is a need to clarify any of the responses.

Participation in this study is voluntary and the participant can abandon the study at any time.

By clicking on "Next", I declare that I am over 18 and voluntarily participate in this study, having read the information contained in this term before participating.

A.2 Participants Profile Form

Evaluation Study

sidornellas@gmail.com Switch account

* Required

Participant Profile

PP1. Name (optional)

Your answer

PP2. Highest Academic degree *

- ◯ Elementary School
- ◯ High School
- ◯ Bachelor's degree
- ◯ Specialization
- ◯ Masters' degree
- ◯ Ph.D. degree

PP3. Situation of the highest academic degree *

- ◯ Complete
- ◯ Incomplete

PP4. Theoretical knowledge of UX evaluation *

- ◯ None
- ◯ Low (i.e., knowledge acquired by yourself through books, videos or other materials)
- ◯ Medium (i.e., acquired mainly during courses or undergraduate research)
- ◯ High (i.e., you are an expert, have a certification or master or doctorate research related to UX evaluation)

PP5. If the theoretical knowledge was obtained in courses, inform which areas were the courses related to

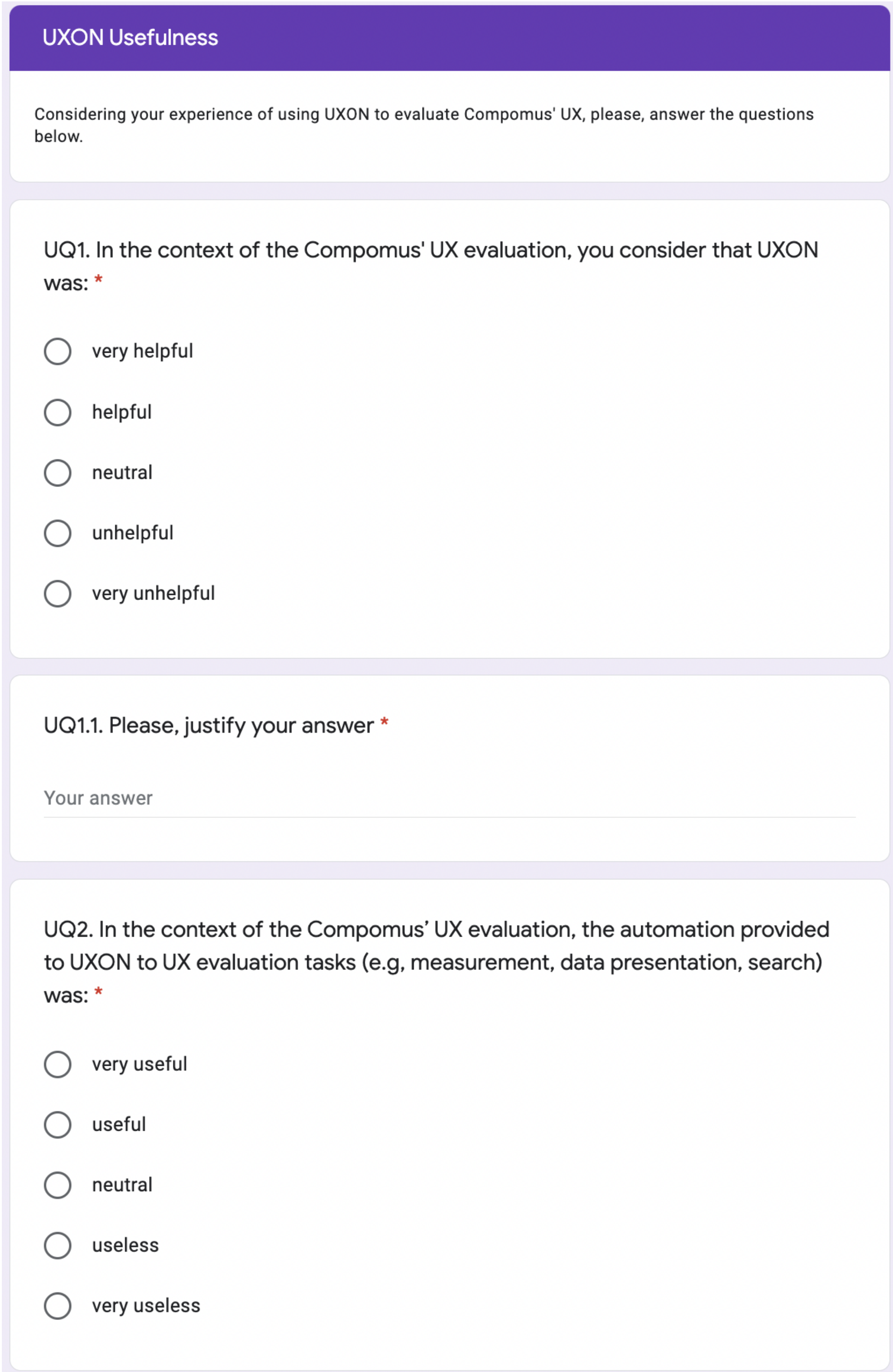
- ◯ Arts
- ◯ Computer Science
- ◯ Social Communication
- ◯ Design
- ◯ Other: ____

PP6. Practical experience with UX Evaluation *

- ◯ None
- ◯ Low (less than 1 year)
- ◯ Medium (between 1 and 3 years)
- ◯ High (more than 3 years)

A.3 Questionnaire

A.3.1 UXON Usefulness



UXON Usefulness

Considering your experience of using UXON to evaluate Compomus' UX, please, answer the questions below.

UQ1. In the context of the Compomus' UX evaluation, you consider that UXON was: *

- ◯ very helpful
- ◯ helpful
- ◯ neutral
- ◯ unhelpful
- ◯ very unhelpful

UQ1.1. Please, justify your answer *

Your answer

UQ2. In the context of the Compomus' UX evaluation, the automation provided to UXON to UX evaluation tasks (e.g, measurement, data presentation, search) was: *

- ◯ very useful
- ◯ useful
- ◯ neutral
- ◯ useless
- ◯ very useless

UQ2.1. Please, justify your answer *

Your answer

UQ3. Regarding data presentation, when compared to the previous solution used to evaluate Compomus' UX, the adoption of UXON: *

- ◯ strongly improved the quality of data presentation
- ◯ improved the quality of data presentation
- ◯ it was neutral
- ◯ worsened the quality of data presentation
- ◯ strongly worsened the quality of data presentation

UQ3.1. Please, justify your answer *

Your answer

UQ4. Regarding data analysis (based on the collected data) and the reached conclusions, when compared to the previous solution used to evaluate Compomus' UX, the adoption of UXON: *

- ◯ strongly improved the quality of data analysis and reached conclusions
- ◯ improved the quality of data analysis and reached conclusions
- ◯ it was neutral
- ◯ worsened the quality of data analysis and reached conclusions
- ◯ strongly worsened the quality of data analysis and reached conclusions

UQ4.1. Please, justify your answer *

Your answer

UQ5. Regarding the identification of improvement needs in the Compomus' UI/UX, when compared to the previous solution used to evaluate Compomus' UX, the adoption of UXON: *

- ◯ strongly improved the identification of improvement needs in the Compomus UI/UX
- ◯ improved the identification of improvement needs in the Compomus UI/UX
- ◯ it was neutral
- ◯ worsened the identification of improvement needs in the Compomus UI/UX
- ◯ strongly worsened the identification of improvement needs in the Compomus UI/UX

UQ5.1. Please, justify your answer *

Your answer

UQ6. Regarding time and effort spent to evaluate Compomus' UX, when compared to the previous solution used to evaluate Compomus' UX, the adoption of UXON: *

- ◯ strongly decreased time and effort
- ◯ decreased time and effort
- ◯ it was neutral
- ◯ increased time and effort
- ◯ strongly increased time and effort

UQ6.1. Please, justify your answer *

Your answer

UQ7. How much does UXON support the following activities during an UX evaluation? *

	Planning (e.g., definition of the metrics to be used)	Data collection (e.g., data capture, formatting and storage)	Data analysis & results (e.g., data visualization, reaching conclusions)
strongly support	○	○	○
support	○	○	○
neutral	○	○	○
does not support	○	○	○
strongly does not support	○	○	○

UQ7.1. Please, justify your answer *

Your answer

A.3.2 UXON Feasibility

UXON Feasibility

Considering your experience of using UXON to evaluate the Compomus' UX, please, answer the questions below.

FQ1. Concerning ease of use, in your opinion UXON is: *

- very easy to use
- easy to use
- neutral
- difficult to use
- very difficult to use

FQ1.1. Please, justify your answer *

Your answer

FQ2. Regarding the terminology used in UXON, you consider it: *

- very consistent with the application domain (UX evaluation)
- consistent with the application domain (UX evaluation)
- neutral
- inconsistent with the application domain (UX evaluation)
- very inconsistent with the application domain (UX evaluation)

FQ2.1. Please, justify your answer *

Your answer

FQ3. Would you use UXON again? *

- ◯ yes
- ◯ no

FQ1.1. Please, justify your answer *

Your answer

FQ4. Would you recommend UXON for other people? *

- ◯ yes
- ◯ no

FQ4.1. Please, justify your answer *

Your answer

A.3.3 UXON General Evaluation and Suggestion

General Evaluation and Suggestions

Considering your experience of using UXON to evaluate the Compomus' UX, please, answer the questions below.

GQ1. What do you think of extending UXON to support UX evaluation of other applications whose evaluation is based on interaction logging data? *

- it would be very useful
- it would be useful
- it wouldn't matter
- it would be useless
- it would be very useless

GQ1.1. Please, justify your answer *

Your answer

GQ2. What are the main advantages of using UXON? *

Your answer

GQ3. What are the main disadvantages of using UXON? *

Your answer

APPENDIX B – Artifacts used in UXON's Evaluation with Developer

B.1 Consent Form

Interview on the development of UXON

This study is being carried out in the context of the doctoral research of the student Simone Dornelas Costa, carried out in the Graduate Program in Informatics of the Federal University of Espírito Santo, under the guidance of professors Monalessa Perini Barcellos and Ricardo de Almeida Falbo (in memoriam).

The aim of the study is to investigate the use of HCI-ON in the development of applications to support the solution of problems related to HCI, from the developer's point of view.

For this, the following research questions are considered:

(RQ1) How does the use of (an extract of) HCI-ON help in the development of applications to support the solution of problems related to HCI?

(RQ2) What are the identified benefits and difficulties of using (an extract of) HCI-ON in application development to support solving HCI-related problems?

Each research question was broken down into more specific questions to be answered by the UXON developer in an interview.

After accepting this consent form, some questions will be presented. The confidentiality of the individual data provided in the study is guaranteed. The data are intended for research purposes and are not used for personal or professional evaluation. Name (optional) and e-mail of the participants are requested only for contact purposes, in case there is a need to clarify any of the answers.

Although invited, participation in the study is voluntary, with the right not to want to participate or to abandon the study at any time.

By clicking on "Next", I declare that I am over 18 years of age and voluntarily participate in this study, having read the information contained in this term before participating.

sidornellas@gmail.com Switch account

* Required

Email *

Your email

B.2 Participants Profile Form

Participant Profile

PP1. Name

Your answer

PP2. Position currently held *

Your answer

PP3. Highest Academic degree *

- ◯ Elementary School
- ◯ High School
- ◯ Bachelor's degree

PP4. Situation of the highest academic degree *

- ◯ Complete
- ◯ Incomplete

PP5. Inform the course or area of the highest degree of academic degree indicated *

- ◯ Computer science
- ◯ Information systems
- ◯ Computer engineering
- ◯ Other: ____

PP6. Theoretical and practical knowledge in systems development, ontologies and ontologies-based systems development: *

	None	Low (did not take any courses or disciplines and gained some knowledge by reading books or other materials)	Medium (did a course or training on the subject, lasting at least 4 hours or worked with the subject in the Scientific Initiation or Graduation Project or in small projects in a company)	High (expert in the subject, have some certification in the area or have worked with the subject in master's or doctoral research or in a company)
Systems development (theoretical)	◯	◯	◯	◯
Systems development (practical)	◯	◯	◯	◯
Ontology (theoretical)	◯	◯	◯	◯
Ontology (practical)	◯	◯	◯	◯
Development of ontology-based systems (theoretical)	◯	◯	◯	◯
Development of ontology-based systems (practical)	◯	◯	◯	◯