Towards an Ontology Network on Human-Computer Interaction

Simone Dornelas Costa[1,2], Monalessa Perini Barcellos[2], Ricardo de Almeida Falbo[2], and Murillo Vasconcelos Henriques Bittencourt Castro[2]

[1] Computer Department, Federal University of Espírito Santo (UFES), Alegre, Brazil
simone.costa@ufes.br

[2] Ontology & Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo (UFES), Vitória, Brazil
{monalessa,falbo}@inf.ufes.br, murillo.castro@aluno.ufes.br

Abstract. Engineering interactive computer systems is a challenging task that involves concerns related to the human-computer interaction (HCI), such as usability and user experience. HCI is a wide domain, where ontologies are useful instruments for supporting knowledge-related problems. However, HCI ontologies have been built and used in isolation. Ideally, in wide domains, ontologies should not be stand-alone artifacts. They should relate to each other, forming a network of interlinked semantic resources, i.e. an ontology network. Therefore, in this paper we introduce HCI-ON, a Human-Computer Interaction Ontology Network composed of ontologies that we have developed and others found in the literature. HCI-ON organizes and integrates knowledge, serving as a basis to several applications. We also discuss mechanisms to evolve HCI-ON and present some envisioned applications.

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

1 Introduction

Developing Interactive Computer Systems (ICS) is a challenging task, which involves a diverse body of knowledge and multidisciplinary teams, joining people from different backgrounds with their own technical language, terms and knowledge [5,4]. To an ICS reach high usability levels, it is necessary to take HCI aspects into account. HCI is a wide domain and as the area matures, new terms are proposed and new meanings are assigned to existing terms. This makes it difficult to establish a common conceptualization about HCI, leading to semantic interoperability problems, such as ambiguity and imprecision when interpreting shared information. Ontologies can be useful to capture and organize knowledge to deal with these problems. They have been applied in the HCI context to knowledge representation [7,20], to aid in interaction design [2,27,34] and evaluation [24], interface adaptation [3,16,25], semantic annotation [13,21], among others.

We investigated the state of the art of ontologies on HCI through a systematic literature review and we found 22 ontologies. However, there are several HCI aspects not covered by them. Since HCI is a complex domain, it is not possible to build a large monolithic ontology to cover the entire domain. Contrariwise, HCI ontologies should be built incrementally in an integrated way, forming a network. An ontology network (ON) is a collection of ontologies related together by means of dependency and alignment relationships [33]. ONs enable to establish a comprehensive conceptualization that provides a common understanding about the domain and can be used as a reference to solve 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 its universe of discourse.

In this paper we argue that HCI ontologies should be organized as an ON, to provide comprehensive knowledge about the domain and support knowledge evolution. We introduce the Human-Computer Interaction Ontology Network (HCI-ON), an ON composed of ontologies that we have developed and others we found in the literature. Since HCI-ON is very extensive, our focus here is to provide a general view of its three-layered architecture and discuss its evolution mechanisms. Section 2 briefly presents the background for the paper. Section 3 presents HCI-ON architecture and the HCI Design Ontology. Section 4 discusses how to evolve HCI-ON by adding new and existing ontologies to it. Finally, Section 5 presents some envisioned applications and our final considerations.

2 Background

HCI involves multidisciplinary knowledge and people from different communities. The lack of a common conceptualization shared by the communities interested in HCI can lead to communication, knowledge transferring and semantic interoperability problems.

Ontologies have been acknowledged as quite appropriate to solve semantic conflicts, for reducing conceptual ambiguities and inconsistencies, for making knowledge structures clearer and can be used for establishing a common conceptualization of a domain of interest. According to [28], ontologies can be organized in a three-layered architecture: (i) *foundational ontologies* model the very basic and general concepts and relations that make up the world (e.g., objects, events); (ii) *core ontologies* refine (i) by adding detailed concepts and relations in a specific area (e.g., service, process); and, (iii) *domain ontologies* describe a particular domain in reality (e.g., the anatomy of the human body) by specializing concepts from (i) or (ii).

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 [33]. 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 [26]. In such cases, building an *ontology network* (ON) is an adequate solution [33]. In an ON, ontologies are connected to each other through relationships, such as *dependency* and *alignment*. The former occurs 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). The latter 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) [33].

To investigate existing ontologies in HCI, we have performed a systematic literature review (SLR) [15] and found 22 ontologies: [31], [32], [2], [3], [6], [18], [29], [34], [7], [9], [13], [21], [27], [20], [35], [11], [19], [1], [17], [16], [24], [25]. These ontologies cover different, but related HCI aspects. Some of them focus on UI (User Interface) design and evaluation, representing aspects related to both the user and the system, but without describing the HCI phenomenon itself ([32], [2], [34], [11], [17], [16]). Others address only one of the parts involved in the HCI phenomenon: user ([21], [27]) or system ([6], [18], [29], [13], [1]). Four ontologies describe the HCI phenomena: [31], [7], [35] and [20]. However, [7] and [20] are specific to some kinds of interaction, namely: haptic ([20]) or by means of gestures ([7]). [3] covers different aspects involving adaptation. [9] is specific to context of use. Three works ([17], [16], [25]) focus on people with disabilities. [19] addresses the characterization of user feedback and [24] focuses on usability guidelines and related elements. Further information about the 22 ontologies and its concepts is available at `http://bit.ly/SLR_OntoInHCI`.

Although a variety of HCI aspects are addressed by these ontologies, there are aspects not covered by any of them (e.g., HCI evaluation and design processes, prototype, among others). Moreover, none of them reused or even discuss how to reuse or integrate ontologies. In fact, HCI ontologies have been developed to solve specific problems in specific contexts, without a concern with integration. This approach has proven to be inadequate to integrate, use and share knowledge [26]. To speed up the development and use of HCI ontologies, we advocate that they should be built incrementally, reusing existing ontologies, and forming a network. This motivated us to work on HCI-ON.

3 HCI-ON: A Human-Computer Interaction Ontology Network

Fig. 1 presents HCI-ON (current version). Since HCI is related to Software Engineering, HCI-ON is integrated to SEON (Software Engineering Ontology Network) [26]. In the figure, each circle (network node) represents a core or a domain ontology. Obfuscated circles represent ontologies under development. Arrows represent dependency relationships, indicating that concepts from the target ontology are reused by the source ontology (in red from HCI-ON to SEON). Circle size varies according to the number of concepts of the ontology.

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

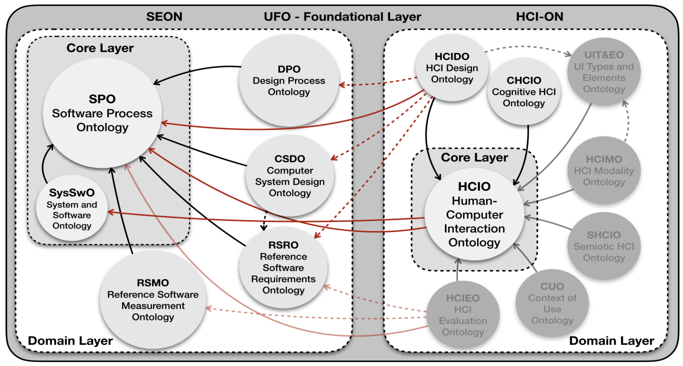

Fig. 1. HCI-ON Architecture.

ontologies into it. In this sense, in HCI-ON, we adopted a three-layered architecture (Fig. 1). At the *foundational layer*, we adopted UFO (Unified Foundational Ontology) [12], which is also used in SEON. By doing that, we keep the same foundation on both ONs concepts, making easier to connect them. At the *core layer*, we developed the Human-Computer Interaction Ontology (HCIO) [8]. 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. Finally, at the *domain layer*, there are domain-specific ontologies, namely: HCI Design Ontology (HCIDO); HCI Evaluation Ontology (HCIEO); UI Types and Elements Ontology (UIT&EO); HCI Modality Ontology (HCIMO); Context of Use Ontology (CUO); Cognitive HCI Ontology (CHCIO); and Semiotic HCI Ontology (SHCIO). 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 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. CHCIO and SHCIO also describe the HCI phenomenon. The former does that by adopting the Seven Stages of Action perspective proposed by [22], while the latter adopts the meta-communication perspective proposed by [30].

The decision on which 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 (from design, UI, modalities of interaction, evaluation to context of use). Moreover, they allow describing the HCI phenomenon under cognitive and semiotic perspectives. The ontologies in the domain layer support the HCI-ON growth, since each of them can be extended to address more specific sub-domains or related domains. Next, we present a fragment of the HCI Design Ontology (HCIDO).

3.1 HCI Design Ontology

The HCI Design Ontology (HCIDO) addresses aspects such as which types of HCI objects can be designed, the involved artifacts and agents that deal with them. HCIDO was developed by specializing concepts mainly from HCIO (HCI-ON) and CSDO (SEON). CSDO deals with the design of computer systems. We built CSDO based mainly on the conceptual model proposed by Ralph and Wand [23], who define design (in general) as "a specification of an object, manifested by an agent, intended to accomplish goals, in a particular environment, using a set of primitive components, satisfying a set of requirements, subject to constraints". To develop HCIDO, we reused CSDO by specializing its concepts and connecting them with specializations of HCIO concepts. By doing that, we address HCI design by connecting design aspects to HCI objects.

Fig. 2 presents an overview of an HCIDO fragment and its dependencies with ontologies from SEON and HCI-ON. The black single-dashed horizontal lines separate concepts from different ontologies, and the red double-dashed lines separate the layers. The top-most is the foundational layer, where concepts from UFO are placed. Following a top-down direction, there are three ontologies at the core layer: SPO and SysSwO (SEON); and HCIO (HCI-ON). They provide details about the agents and objects involved in SE and HCI and relationships between them. At the bottom, there is the domain layer, where HCIDO is located, directly connected to CSDO (SEON), UIT&EO and HCIO. Different colors are used to indicate concepts from different ontologies. Next, we describe HCIDO concepts. In the text, concepts are written in **bold** and examples (i.e., instances) in *italics*.

From HCIO, there are three concepts that characterize possible objects of interest in HCI Design, all of them are **Software Items**, i.e., pieces of software produced in software processes [10]. The first, **User Interface Program**, represents software items that aim at producing a certain result through execution on a computer, dealing with **User Interface**, which is composed of **Input Equipment** and **Output Equipment** (not shown in Fig. 2). The second, **Interactive Software System**, is a software item constituted by at least one **User Interface Program** and, being a Software System, it intends to determine the behavior of the computer towards the external environment [10]. The third, **Interactive Computer System**, is a computer system that has **User Interface**. It is a combination of hardware and software used to process, transform, store, display or transmit information or data by receiving input, and communicating output to users [14]. For example, *Microsoft Visual Studio (MVS)* is an instance of **Interactive Software System**. Among the many programs that constitute it, the ones that deal with its graphical interface are instances of **User Interface Program**. The *MVS* loaded and running on a computer, together with

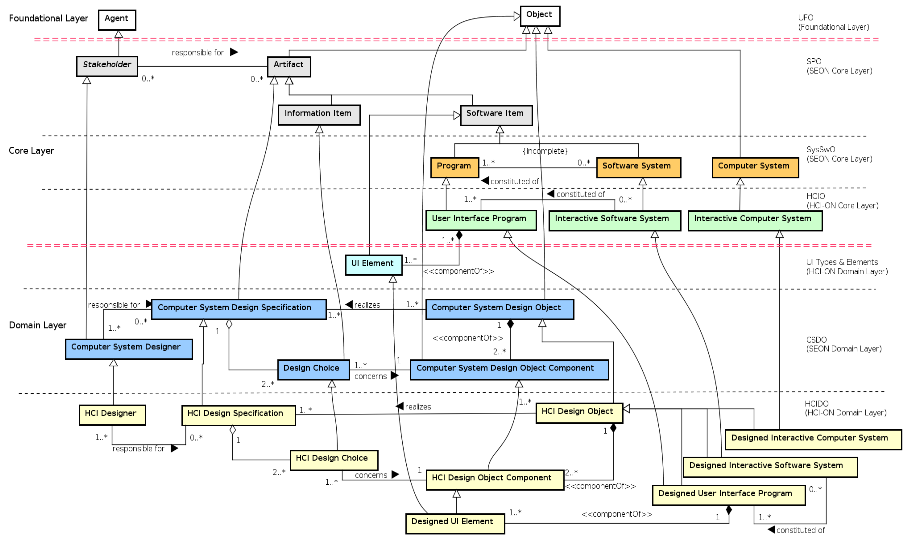

Fig. 2. HCIDO and related ontologies.

input and output equipment to interact with the user, comprises an instance of **Interactive Computer System**.

We consider that, in the context of HCI, **User Interface Programs**, **Interactive Software Systems** (including their **User Interfaces**) and **Interactive Computer Systems** (involving software and hardware components) are the kind of objects that can be designed. Therefore, in HCIDO, we define **HCI Design Object** as the designed objects in the HCI phenomenon, being either a **Designed User Interface Program**, a **Designed Interactive Software System** or a **Designed Interactive Computer System**. What adds the "designed" characteristic to these objects is the existence of an **Artifact** with a detailed description of them in terms of their design (e.g., a prototype or a document describing their components and connections among them). This artifact is an **HCI Design Specification**, which consists of a collection of several **Information Items**, named **HCI Design Choices**, each one concerning how a specific **HCI Design Object Component** should be. **HCI Design Object Components** are smaller parts that, connected, form an **HCI Design Object**, which thus realizes the **HCI Design Specification** where its components are described. **Designed UI Elements** are **UI Elements** (e.g. windows, buttons, toolbars) from UIT&EO that compose a **Designed User Interface Program**. Finally, the **HCI Designer** is the agent (a person or a group) responsible for creating and maintaining the **HCI Design Specification**.

Taking the *MVS* example, in an HCI design scenario, it is a **Designed Interactive Software System**. The team or person in charge of designing it (i.e., the **HCI Designer**) sketched a prototype (**HCI Design Specification**) showing how *MVS* graphical user interface should be. This prototype encoded several **HCI Design Choices**, like the description of how a toolbar (a **Designed UI Element**) should look and be positioned on the screen. This toolbar is implemented as an **HCI Design Object Component** in the *MVS* graphical user interface program (**Designed User Interface Program**), which is a constituent of the **Designed Interactive Software System** (*MVS*). What makes the *MVS* an **HCI Design Object** is the fact that it has a design specification describing its characteristics (i.e., an **HCI Design Specification**) and once it was created satisfying that specification, it realizes that.

Although not shown in Fig. 2, HCIDO also addresses concepts related to the mental aspects involved in HCI design. For example, both the design object and its specification exist in the mind of the designer before being materialized as the objects and artifacts showed in Fig. 2. Moreover, as it can be seen in Fig. 1, HCIDO connects to RSRO. This relation allows to align the HCI Design Specification to the functional (e.g., functionalities that the software should provide) and non-functional (e.g., usability requirements to be satisfied) requirements that must be met by the software item. It makes explicit that the HCI Design Specification must describe HCI Design Choices able to meet the software item requirements, connecting the HCI Design process to the Requirements Engineering process.

4 Evolving HCI-ON

An ON is constantly evolving. Each ontology added to ON contributes for it as a whole. 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.

To evolve HCI-ON, one can (i) develop new ontologies from scratch and add them to the network; or (ii) add existing ontologies to the network. In (i), the ontologies must be developed grounded in UFO, to share the same foundation of all networked ontologies, and they must be added to the network through dependency relationships. We have developed the HCI-ON ontologies shown in Fig. 1 by following this procedure. In (ii), one can use dependency or alignment relationships. In the first case, it is necessary to re-engineer the existing ontology in the light of UFO, so that the ontology will share the same HCI-ON basic conceptualization and, thus, it will be possible to integrate it into the network properly. When two or more existing ontologies addressing the same subject represent together the conceptualization of a new ontology to be added to HCI-ON (i.e., they are complementary), they must be merged and re-engineered in the light of UFO. After re-engineering, the ontology can be added to HCI-ON through dependency relationships. For example, we can merge and re-engineer the ontologies [21] and [27] to produce a Persona Ontology; [3], [21], [17], [16] and [25] to produce a User Capacity and Accessibility Ontology and add them to HCI-ON.

Existing ontologies can also be added to the ON as they are, through alignment relationships (i.e., indicating equivalence between concepts of different on-

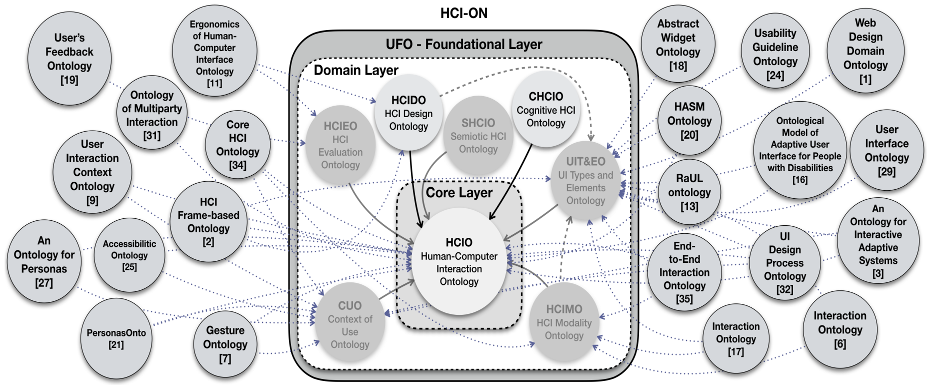

Fig. 3. Evolving HCI-ON.

tologies). For example, the terms People ([31]), Person ([21]) and Person ([27]) have the same meaning than HCIO's Person concept. Aligning existing ontologies to HCI-ON makes the ON conceptualization more comprehensive. Moreover, it allows the plugged ontologies not to be changed (not affecting applications in which they are used) and, even so, extend their conceptualizations. Fig. 3 shows the ontologies we found in our SLR plugged into HCI-ON. Since they were not re-engineered, they are not grounded in UFO.

5 Final Considerations

In this paper, we advocate that HCI ontologies should be built forming a network that organizes and structures knowledge. This motivated us to create HCI-ON, which aims to provide a comprehensive conceptualization about HCI.

We envisioned some applications to HCI-ON. First, it can be useful to solve knowledge and semantics-related problems. For example, it can be used for communication purposes, to support 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. HCI-ON can also be used to annotate HCI-related documents (e.g., text document, spreadsheets, images), allowing easily to retrieve and integrate information from these documents. 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. HCI-ON can also aid in systems integration. In integration scenarios spanning different HCI subdomains, the benefits of using HCI-ON stand out. Instead of spending effort to integrate several ontologies, one can just extract the HCI-ON portion to be used. Another application concerns semantic interoperability among standards. Considering that different standards often presents a diverse vocabulary leading to semantic conflicts, HCI-ON can serve as the reference conceptualization to harmonize them, so that they share the same conceptualization and can be properly used in a combined way.

We have experienced the benefits of ONs by using SEON in applications as the ones we cited before. However, when talking about ICS, HCI conceptualization is also necessary. Therefore, we intend to use HCI-ON in these applications and explore it to provide solutions integrating HCI aspects to SE practices, by connecting HCI designers and software engineers. In this sense, currently, we are working on a knowledge-based solution to HCI design and evaluation.

References

1. Bakaev, M., Gaedke, M.: Application of evolutionary algorithms in interaction design: From requirements and ontology to optimized web interface. In: NW Russia Young Researchers in Electrical and Electronic Eng. Conf. pp. 129–134 (2016)
2. Bakaev, M., Avdeenko, T.: User interface design guidelines arrangement in a recommender system with frame ontology. In: Database Systems for Advanced Applications. pp. 311–322. Springer (2012)
3. Bezold, M., Minker, W.: A Framework for Adapting Interactive Systems to User Behavior. Journal of Ambient Intelligence and Smart Env. **2**(4), 369–387 (2010)
4. Carroll, J.M.: Human computer interaction (hci). In: Soegaard, M., Dam, R.F. (eds.) The Ency of HCI, chap. 2, pp. 21–61. The Interaction Design Foundation, 2nd edn. (2014)
5. Castro, M.V.H.B., Barcellos, M.P., Falbo, R.d.A., Costa, S.D.: Knowledge Management in Human-Computer Interaction Design: A Mapping Study, in press
6. Celino, I., Corcoglioniti, F.: Towards the Formalization of Interaction Semantics. In: Proc. of the Int. Conf. on Semantic Systems. pp. 10:1–10:8 (2010)
7. Chera, C., Tsai, W., Vatavu, R.: Gesture ontology for informing service-oriented architecture. In: Int. Symposium on Intelligent Control. pp. 1184–1189. IEEE (2012)
8. Costa, S.D., Barcellos, M.P., Falbo, R.d.A.: What human-computer interaction is: An ontological view, under review
9. Devaurs, D., Rath, A., Lindstaedt, S.: Exploiting the user interaction context for automatic task detection. Applied Artificial Intelligence **26**, 58–80 (2012)
10. Duarte, B.B., de Castro Leal, A.L., Falbo, R.d.A., Guizzardi, G., Guizzardi, R.S.S., Souza, V.E.S.: Ontological foundations for software requirements with a focus on requirements at runtime. Applied Ontology **13**(2), 73–105 (2018)
11. Elyusufi, Y., Seghiouer, H., Alimam, M.A.: Building profiles based on ontology for recommendation custom interfaces. In: Int. Conf. on Multimedia Computing and Systems. pp. 558–562 (2014)
12. Guizzardi, G.: Ontological foundations for structural conceptual models. Ph.D. thesis, University of Twente (2005)
13. Haller, A., Groza, T., Rosenberg, F.: Interacting with Linked Data via Semantically Annotated Widgets. In: Int. Semantic Tech Conf. pp. 300–317. Springer (2012)
14. ISO 9241-210: Iso 9241-210:2019(en) - ergonomics of human-system interaction - part 210: Human-centred design for interactive systems (2019)
15. Kitchenham, B.A., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Kele Univ. and Univ. of Durham (2007)
16. Kultsova, M., Potseluico, A., Zhukova, I., Skorikov, A., Romanenko, R.: A two-phase method of user interface adaptation for people with special needs. In: Creativity in Intelligent Tech and Data Science. pp. 805–821. Springer (2017)

17. Lebib, F.Z., Mellah, H., Mohand-Oussaid, L.: Ontological interaction modeling and semantic rule-based reasoning for user interface adaptation. In: Proc. of the Int. Conf. on Web Info. Systems and Tech. pp. 347–354. INSTICC, SciTePress (2016)
18. Martín, A., Rossi, G., Cechich, A., Gordillo, S.: Engineering accessible web applications. an aspect-oriented approach. World Wide Web **13**(4), 419–440 (2010)
19. Mezhoudi, N., Vanderdonckt, J.: A user's feedback ontology for context-aware interaction. In: World Symp on Web Applications and Networking. pp. 1–7. IEEE (2015)
20. 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 **40**(4), 1358–1371 (2013)
21. Negru, S., Buraga, S.: A knowledge-based approach to the user-centered design process. In: Int. Joint Conf. on Knowledge Discovery, Knowledge Eng. and Knowledge Management. pp. 165–178. Springer (2013)
22. Norman, D.A.: The Design of Everyday Things: Revised and Expanded Edition. Basic Books, 2nd edn. (2013)
23. Ralph, P., Wand, Y.: A proposal for a formal definition of the design concept. In: Design Requirements Eng.: A Ten-Year Perspective. pp. 103–136. Springer (2009)
24. Robal, T., Marenkov, J., Kalja, A.: Ontology Design for Automatic Evaluation of Web User Interface Usability. In: Portland Int. Conf. on Management of Eng. and Tech. pp. 1–8. IEEE (2017)
25. Romero Mariño, B., Rodríguez Fórtiz, M.J., Hurtado, M., M. Haddad, H.: Accessibility and Activity-Centered Design for ICT Users: ACCESIBILITIC Ontology. IEEE Access **6**, 60655–60665 (2018)
26. Ruy, F.B., Falbo, R.d.A., Barcellos, M.P., Costa, S.D., Guizzardi, G.: Seon: A software engineering ontology network. In: Knowledge Eng. and Knowledge Management. pp. 527–542. Springer (2016)
27. Salma, G., Marouane, E.L.: Operating an application for modeling persona by using ontologies. Journal of Theoretical & Applied Info. Tech **88**(1) (2016)
28. Scherp, A., Saathoff, C., Franz, T., Staab, S.: Designing core ontologies. Applied Ontology **6**(3), 177–221 (2011)
29. Shahzad, S., Granitzer, M., Helic, D.: Ontological model driven gui development: User interface ontology approach. In: Int. Conf. on Computer Sciences and Convergence Info. Tech. pp. 214–218. IEEE (2011)
30. Souza, C.S.d.: The Semiotic Engineering of Human-Computer Interaction. The MIT Press (2005)
31. Storrs, G.: A conceptualisation of multiparty interaction. Interacting with Computers **6**(2), 173–189 (1994)
32. Suàrez, P.R., Jùnior, B.L., De Barros, M.A.: Applying knowledge management in ui design process. In: Proc. of Conf. on Task Models and Diagrams. pp. 113–120. ACM (2004)
33. Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A.: Ontology Engineering in a Networked World. Springer (2012)
34. Tourwé, T., Tsipokova, E., González-Deleito, N., Hristoskova, A.: Ontology-driven elicitation of multimodal user interface design recommendations. In: Belgian/Netherlands Artificial Intelligence Conf. p. 8 (2011)
35. Zamzami, E.M., Budiardjo, E.K., Suhartanto, H.: Requirements recovery using ontology model for capturing end-to-end interaction of proven application software. Int. Journal of Software Eng. and its Applications **7**(6), 425–434 (2013)