Towards an Ontology-based Approach to Develop Software Systems with Adaptive User Interface

Alexandre A. C. de Freitas
Ontology and Conceptual Modeling Research Group, Federal University of Espírito Santo
Vitória, ES, Brazil
alexandre.a.freitas@edu.ufes.br

Murilo B. Scalser
Ontology and Conceptual Modeling Research Group, Federal University of Espírito Santo
Vitória, ES, Brazil
muriloborghardt7@gmail.com

Simone D. Costa
Computer Science Department, Federal University of Espírito Santo
Alegre, ES, Brazil
simone.costa@ufes.br

Monalessa P. Barcellos
Ontology and Conceptual Modeling Research Group, Federal University of Espírito Santo
Vitória, ES, Brazil
monalessa@inf.ufes.br

ABSTRACT

The new ways of manipulating computers, smartphones and other devices have brought challenges such as the need to ensure a good usability when different user types use the same system. Adaptive user interface (AUI) systems are a possible solution. They change the user interface to better meet the needs of different users. However, developing such systems is not trivial. It is necessary to capture the users' characteristics and preferences and constantly adapt the system accordingly. In this paper, we discuss the use of ontologies to support the development of AUI systems. We argue that by providing structured knowledge about such systems, ontologies help understand how they work and offer a basis to structure them, identify the necessary adaptations and implement mechanisms to make them happen in run-time. We have explored the use of ontologies from an ontology network to develop a social network about academic subjects that automatically adapts its interface according to the low vision and colorblind user's needs and usage characteristics. The first version of an ontology-based process to guide the development of AUI systems raised from this experience.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Interactive systems and tools**.

KEYWORDS

Adaptive User Interface, Ontology, Ontology Network

ACM Reference Format:
Alexandre A. C. de Freitas, Murilo B. Scalser, Simone D. Costa, and Monalessa P. Barcellos. 2022. Towards an Ontology-based Approach to Develop Software Systems with Adaptive User Interface. In *XXI Brazilian Symposium on Human Factors in Computing Systems (IHC '22), October 17–21, 2022, Diamantina, Brazil.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3554364.3559139

IHC '22, October 17–21, 2022, Diamantina, Brazil

ACM ISBN 978-1-4503-9506-9/22/10…$15.00
https://doi.org/10.1145/3554364.3559139

1 INTRODUCTION

Nowadays, the digital and contemporaneous society has required interactive systems even more intuitive and suitable for the user needs. For that, a proper user interface (UI) is needed. Efficient UIs promote an effective communication between users and the system. Thus, well-designed UI is essential for the success of any interactive system. However, the increasing number of different electronic devices, environments and types of users have been a challenge when developing interactive systems [21].

A well-known and currently discussed issue in this context concerns usability problems when different types of users use the same system [20]. Users differ in a wide range of variables, including demographic characteristics, background, education, personality, cognitive skills and preferences. Users' motivation, goals and moods also vary. In fact, one of the major challenges in the study of human–computer interaction is how to deal effectively with individual differences, preferences, and experience [18].

For a UI to facilitate effective communication between the user and the system, it should consider different needs of different users. The difference can take into account characteristics, such as the ones aforementioned, accessibility needs, and others. Thus, it is important that the UI be able to adapt to the requirements of different types of users [21], i.e., we need adaptive user interfaces (AUI).

AUIs are part of adaptive systems, which are systems that can change, in an automatic way, aspects of their structure or functionality to meet the different needs of users and their changes when using the system [2]. UI adaptations are among the ones that can be performed in an adaptive system. They require identification and classification of user characteristics to constantly adapt the system accordingly [1]. In this paper, we adopt the term *AUI system* referring to such systems, i.e. a system that adapts its UI.

Developing AUI systems is a complex and knowledge-intensive activity [24]. UI adaptations may occur according to a diversity of user information. Hence, structuring and organizing knowledge about the user and the system to promote the proper adaptations in the UI becomes necessary. In this work, we argue that using ontologies is a promising approach to aid in this matter. Ontologies capture and organize knowledge and, thus, can be used to structure knowledge about the interactive system and the users' characteristics, helping understand how such systems work and serving as a basis to structure them, identify the necessary adaptations and implement mechanisms to make them happen in run-time.

In the literature, some works have explored the use of ontologies to develop AUI systems [9] (e.g., [12] and [4]). However, the ontologies often are very specific, i.e., they can only be used to solve a particular problem in the context of the system to each they were created, and are used mainly at operational level. This may work for isolated solutions, but systems have been required to be more comprehensive and constantly evolve according to the user needs. Isolated solutions are usually hard to be extended to incorporate new requirements or reused in the development of new solutions.

Therefore, we advocate that ontologies should also be used at conceptual level to structure knowledge about the system and user characteristics. Thus, it is possible to provide a general knowledge framework that can be used as a basis to define UI adaptations and develop AUI systems. We also argue that, ideally, we should use ontologies from an ontology network (ON), i.e., a set of interconnected ontologies that provides a comprehensive conceptualization about the domain of interest (here, AUI systems). By doing so, it is possible to constantly evolve the set of possible adaptations by considering different concepts from the networked ontologies.

Aiming to explore the use of ontologies from an ON to aid in developing AUI systems, we performed an exploratory study where we used an ontology addressing interactive systems and user profile to develop SNOPI (Social Network with Ontology-based Adaptive Interface), a social network about academic subjects that automatically adapts its UI according to the low vision and colorblind user's needs. The used ontology is a fragment of the Human-Computer Interaction Ontology Network (HCI-ON)[1] [10, 11], an ON that contains several ontologies addressing HCI subdomains. The ontology was useful at the conceptual level by serving as a basis to define the system structural model and at the operational level by providing the semantics used in a reasoning engine to adapt the UI at run-time. From this experience, the first version of an ontology-based process to guide the development of AUI systems emerged.

This paper aims to share the main results of our experience, present the process defined from it, and shine a light on the opportunity of exploring ontologies from an ON to aid in the development of AUI systems. Section 2 provides the background for the paper and discusses some related work; Section 3 briefly introduces SNOPI; Section 4 describes the ontology-based process to develop AUI systems; and Section 5 concludes the paper.

2 BACKGROUND

2.1 Adaptive User Interfaces

Interactive systems are systems that focus on usability for the user [7]. Thus, they have UI. They can also be adaptive systems, i.e., systems able to change features and functionalities automatically, according to the users' needs [19]. It is possible to focus the adaptation specifically on the UI, resulting in an AUI. By considering the different characteristics of different users, the UI will shape itself to meet the user's needs, making the system more accessible and improving its usability [19]. Thus, a UI is adaptive when it automatically adapts itself according to the user's characteristics.

Adaptable UI, which also refers to UI that undergoes adaptations, can be confused with AUI. However, they are distinct concepts. A UI is adaptable when the user can explicitly and deliberately adapt the UI. On the other hand, a UI is adaptive when it adapts automatically to the user's needs [17].

Developing an AUI has some challenges, whether in collecting information from the user, measuring user characteristics, or organizing knowledge through the information collected. Also, there are issues related to the protection and privacy of the user's data, the user's adaptability to a new interface, and the difficulty in adapting the conceptual model of the UI to achieve the desired goals [19].

2.2 Ontologies and Ontology-based AUI

An ontology is a formal and explicit specification of a shared conceptualization [23]. The conceptualization is an abstract and simplified view of the world. In Computer Science, we refer to an ontology as a special kind of information object (a conceptual model), called *reference ontology*, or as a computational artifact, called *operational ontology* [15, 16]. A reference ontology aims to make the best possible description of the domain in reality, regardless of its computational properties. Operational ontologies, in turn, are designed with the focus on guaranteeing desirable computational properties and, thus, are machine-readable ontologies.

For large and complex domains (such as HCI), 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 [5].

Ontologies can aid in software development. Ontology-oriented software development can use both reference and operational ontologies. The former is suitable for supporting the description of the application domain itself and is applied in development time, a.k.a, ontology-driven development (ODD) [22]. The latter is appropriate for use as primary artifacts in run-time and plays a major role in application logic, a.k.a, ontology-based architectures (OBA) [22].

In the literature, there are some works that propose the use of ontologies in the development of AUI systems [9]. For example, de Araújo et al. [12] use a recoloring ontology to develop a functional web prototype that changes the colors of UI elements automatically for colorblind users. The ontology represents knowledge about color vision deficiency types, colors and adaptation techniques, among others. Braham et al. [4], in turn, use ontologies and UI design patterns to develop a mobile application that supports run-time adaptation of the UI for people with disabilities. These works use operational ontologies and did not follow a systematic process to develop the systems, which make it difficult for other people to repeat the process to develop other systems.

In this work, like [12] and [4], we propose to use ontologies to help develop AUI systems (specifically their software constituent). However, our proposal has some important differences. First, we argue for the use of well-founded reference ontologies, which are application-independent and, thus, can be used to develop different AUIs and different systems. Moreover, they can be translated into operational ontologies to be used at run-time. Second, we propose the use of ontologies of an ON. Thus, different ON extracts (i.e.,

[1] https://dev.nemo.inf.ufes.br/hcion/

ontologies containing different concepts) can be used to develop different systems. In addition, the set of user characteristics and other concepts represented in the ON increases over time (because the ON continuously evolves) allowing to address new adaptions. Finally, we are moving towards defining a systematic process that describes the steps to be followed to use ontologies from an ON to develop AUI systems. As a benefit, third parties will be able to use the proposed process to develop AUI systems.

3 USING AN ONTOLOGY EXTRACTED FROM AN ON TO DEVELOP AN AUI SYSTEM

We performed an exploratory study with the purpose of verifying whether the use of an ontology extracted from an ON would support (and how) the development of an AUI system. Since the beginning of the COVID pandemic, many scientific conferences have been held online, making it possible for many people to participate. Information is usually spread in many channels and people can miss it. Thus, we decided to develop a social network devoted to information about scientific events (dates, call for papers, links, publications, etc.). The system, called SNOPI (Social Network with Ontology-based Adaptive Interface), should present the basic features of a social network (e.g., feed, add comments, upvote, downvote) and cover specificities of the application domain. We defined some Personas [8] to identify the main user profiles and needs. In this context, we included Personas with some special needs related to low vision and colorblind.

Once we identified the system requirements, users and their characteristics, we selected an extract of HCI-ON covering aspects related to interactive system, UI, user characteristics, user profile and adaptations. Figure 1 shows a fragment of the ON extract used to develop SNOPI (blue circles show the concepts used in this paper). It includes concepts from five ontologies: *Human-Computer Interaction Ontology* (HCIO), which addresses what an interactive computer system is, user actions taken in an interaction, and how an interaction happens [11]; *User Characterization Ontology* (UCO), which involves concepts related to user characteristics; *UI Types and Elements Ontology* (UIT&EO), which addresses UI types and components; *Adaptive Interface Ontology* (AIO), which deals with AUI and customizations in its components; and *User Profile Ontology* (UPO), which addresses general aspects of a user profile.

Briefly, an *Interactive Computer System* is a computer system that has a *User Interface* composed of *Output Equipment* and *Input Equipment*. An *Adaptive User Interface*, in turn, is a *User Interface* that adapts itself and, thus, it is part of an *Adaptive Interactive Computer System*. *User* participates in *Human-Computer Interactions* to communicate with the system. *User* has *User Characteristics* (e.g., Mary - a user - has 18 years) that are considered to define the *User Profile*, which is composed of several properties (e.g., *User Gender*, *User Disability*, *User Age*, *User Education*, *User Language*, *User Experience Level*, among others). Each property can be specialized into others (e.g., *Cognitive Disability*, *Motor Disability*, *Auditory Disability* and *Vision Disability* are different types of *User Disability*). *User Profiles* are defined by *Adaptive Interactive Computer Systems*. A *User Interface Customization* refers to an adaptation to be made in the *Adaptive User Interface* based on a *User Profile* (the minimum cardinality of this relation is 0 because a customization can be based on other factors such as use context – not represented in Figure 1). A *User Interface Customization* is performed by an *Interface Component Program* (which is a program that composes the *Adaptive Interactive Computer System*) and changes the *Adaptive User Interface*.

SNOPI[2] development followed ODD and OBA approaches. Figure 1 depicts a fragment of the ontology used as a reference conceptual model at dev-time to structure the system and its relational database. The reference ontology also served as a knowledge body that helped better understand AUI systems and related concepts. Moreover, it was transcribed into a language that allows machine-reading, resulting in an operational ontology (ontoSNOPI[3]) used at run-time to support UI adaptation through reasoning.

ontoSNOPI was implemented using OWL[4], and encapsulates information (rules), by means of axioms, on how to adapt the UI according to the user's profile (e.g., considering degrees of color blindness and low vision). The axioms (rules) were defined considering the recommendations of the W3C Accessibility standard [5] and the Web Content Accessibility Guidelines (WCAG)[6]. Shortly, we used the reference ontology to identify the different properties related to *Vision Disabilities* to be considered in the system (i.e., *Blindness* and five types of *Low Vision*, namely: *Field of Vision*, *Light Sensitivity*, *Visual aAcuity (clarity)*, *Color Vision*, *Contrast Sensitivity*) and the aforementioned standards to help identify the adequate adaptations. After defining the adaptation rules (i.e., *User Interface Customizations*), by following the ontology conceptualization, we created *Interface Component Programs* that implement such rules and materialize the adaptation in the UI.

Figure 2 presents an overview of SNOPI. The UI layer contains the components (e.g., screens, controllers) that communicate user and system. The application layer regards the system functionalities, while the data layer concerns data structure and storage. Finally, the semantic layer concerns a reasoning engine that uses the operational ontology and adaptation rules to change the interface according to the user characteristics.

When accessing the system, the user is asked to answer a questionnaire (a partial view is illustrated in Figure 3). The purpose is to capture some user characteristics to support UI adaptation. User data is stored in the database. At run-time, this data is instantiated in the ontoSNOPI and inferences (reasoning) using the axioms are performed from these instances to identify the adaptations suitable for the user profile. The UI adaptations are recorded in the database and the system adapts the UI accordingly. This procedure is performed at the first time the user accesses the system because it does not know the user yet. In the next access, the system checks the captured user information and adapts the UI. At any time, the user can update his/her characteristics in the system. Moreover, he/she can also make new changes in the UI by himself/herself (the UI is adaptive and adaptable).

Figure 4 illustrates the adaptation of the UI to the dark mode (low light emission) and increased font size. This adaptation is suitable for colorblind or light-sensitive user and users with impaired

[2]SNOPI is available at https://dev.nemo.inf.ufes.br/snopi/
[3]ontoSNOPI is available at https://dev.nemo.inf.ufes.br/hcion/ontoSNOPI.owl
[4]https://www.w3.org/TR/owl-guide/
[5]https://www.w3.org/standards/webdesign/accessibility
[6]https://www.w3.org/TR/WCAG/

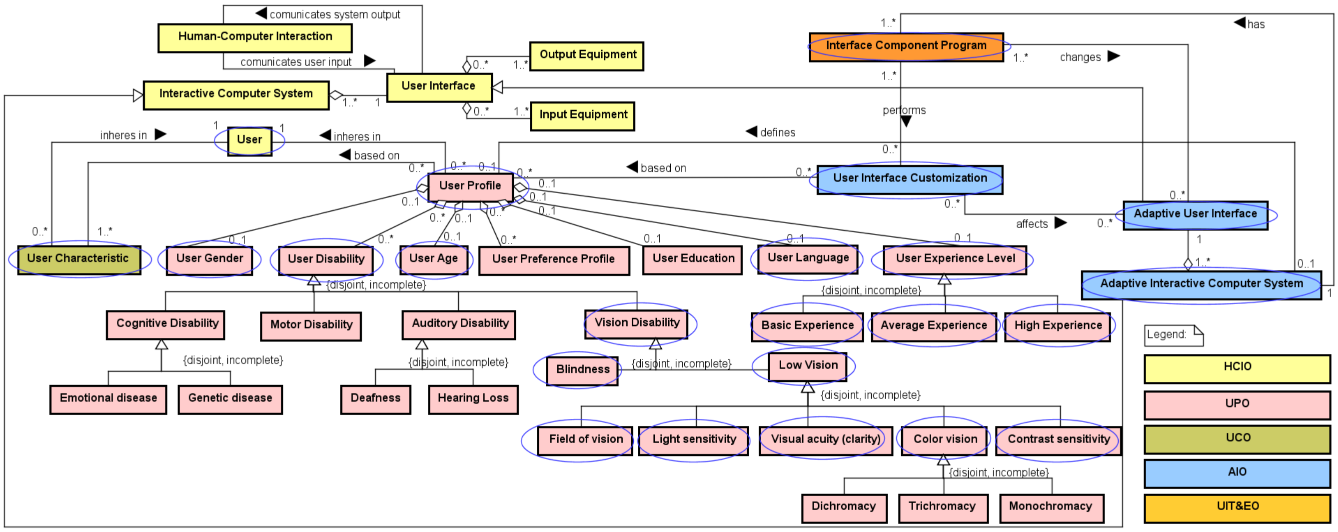

Figure 1: Fragment of the HCI-ON extract used to develop SNOPI.

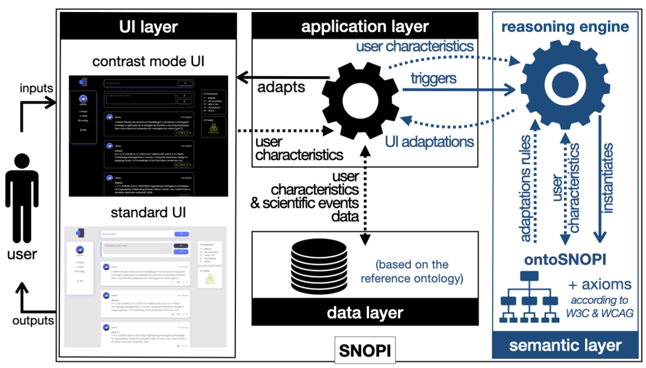

Figure 2: SNOPI overview.

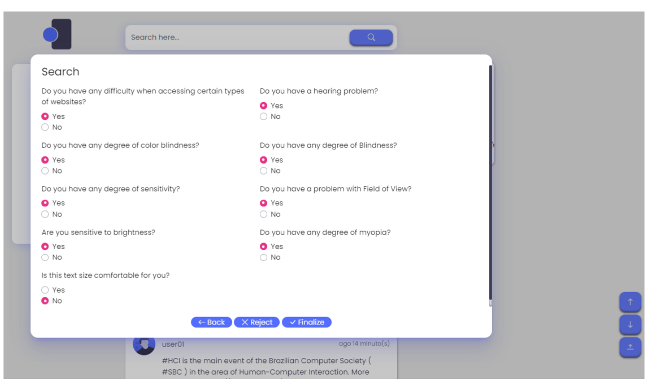

Figure 3: Fragment of the User Characterization questionnaire.

vision. SNOPI performs other UI adaptations, such as to the high contrast mode (Figure 2: *contrast mode UI*), which increases contrast, decreases visual interference, and increases the focus on the components, being suitable for contrast sensitive users.

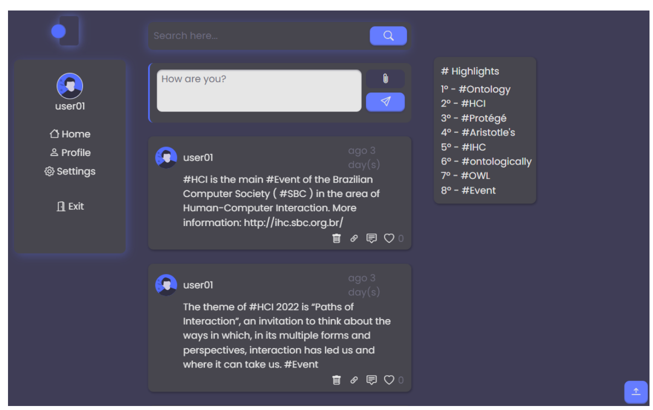

Figure 4: Feed UI adapted for dark mode and large font size.

Developing SNOPI showed us that using ontologies from an ON to support the development of a AUI system is feasible and that the ontology is useful in this matter. The reference ontology supported understanding the interactive system and AUI domain, which helped define how the system should work. In adittion, it aided in identifying user properties that could be considered by the system to define different profiles and served as a basis for the system structural model (we used the ontology model to create the class diagram and define the database structure). Moreover, the operational ontology helped capture user characteristics and change the UI at run-time. Another advantage of using an ON is that we did not need to look for a particular ontology able to address our application problem or develop a new one with that purpose. We just looked at the ON and identified the extract that fit our need.

Based on the SNOPI development experience, we structured the acquired knowledge and lessons learned (i.e., tacit knowledge) by means of a process (i.e., explicit knowledge) that contains the steps to use ontologies from an ON to develop a AUI system.

4 AN ONTOLOGY-BASED PROCESS TO DEVELOP AUI SYSTEMS

In this work, we argue that using an ON makes it easier to apply ontologies to develop AUI systems because the ontology can be extracted from the ON instead of being developed from scratch. As a result, the effort to produce the used ontology decreases. Moreover, the quality of the used ontology increases because ontologies integrated into the ON are properly verified and validated [3]. Thus, the ON is a key pilar in our approach.

In the last years, we have worked on HCI-ON [10, 11], an ON that addresses several HCI subdomains. In this context, recently, we have dedicated efforts to develop ontologies devoted to adaptive interactive computer systems and AUI (a fragment was shown in Figure 1). The ON serves as a knowledge framework that can be used to solve knowledge-related and interoperability problems [10, 11]. In this work, the knowledge-related problem refers to understanding interactive computer systems and AUI subdomains and applying the conceptualization to develop AUI systems. Besides providing knowledge about these subdomains, it is also needed to provide guidelines on how to use the ON to develop AUI systems. Thus, our approach, which we call OADAPT (Ontology-based Approach to Develop AdaPtive inTerfaces), consists of a knowledge framework (i.e., an ON containing ontologies that address aspects relevant to adaptive systems and AUI) and a process describing the steps to use it to develop AUI systems. The process raised from the SNOPI development experience and considers the use of ontologies at dev-time (ODD) and run-time (OBA). Since it is a software development process, it also includes some classic software development phases. Figure 5 shows an overview of the eight-step process to use an ON to develop AUI systems. Next, we briefly describe each step.

(i) Identify System Scope and Users: consists in delimiting the system scope, identifying high-level requirements, the system users and its characteristics. It focuses on understanding the system purpose and boundaries, the problem to be addressed and the expected different types of users. Knowing the users characteristics will help define the adaptations needed in the AUI in next steps. The main user needs should also be identified to establish a initial set of functionalities based on high-level requirements. The adoption of techniques such as Empathy Map [14] and Personas [8] is helpful to identify the users, their needs and characteristics.

(ii) Elicity System Requirements: in this step, the results of the previous one are refined by defining the system functional and non-functional requirements. Functional requirements describe the functions the system should contain. Non-functional requirements, in turn, represent constraints that the system should address. In this sense, non-functional requirements are particularly important to indicate UI adaptation needs. For example, if the system users need UI accessibility options, non-functional requirements specifying such needs should be defined to be further addressed in the UI. Low-fidelity prototypes can be helpful in this step to capture information about features and non-functional requirements.

(iii) Select Ontology: in this step, considering the system scope, requirements and user characteristics, the ON fragment (i.e., the reference ontology) to be used must be selected. If necessary, new concepts can be added to the selected fragment. The reference ontology can be used to refine the results produced in (i) and (ii). For example, the ontology can revel new user characteristics to be considered and can refine non-functional requirements.

(iv) Perform System Analysis: comprises developing the system structural and behavioral models. The reference ontology is used as a basis to the structural model (e.g., class diagram). If necessary, the model can be adjusted to be more suitable for the system (e.g., some concepts can be turned into attributes of other concepts). Guidelines on how to turn ontological models into information models (more adequate to structure systems) can be found in [6].

(v) Define System Architecture: consists in defining the system architecture, its components (e.g., problem domain component, UI component) and related technologies. When adopting OBA, the architecture must include components that portray the use of the operational ontology. The role of the operational ontology must be clearly defined (e.g., its use in the semantic reasoning engine). Moreover, when selecting the technologies, the language and technologies to implement the operational ontology should also be defined.

(vi) Define UI Adaptations: in this step, the UI adaptation rules are defined. Thus, considering the reference ontology, the system architecture (mainly the UI component) and the user characteristics, the UI adaptations to be carried out must be defined. For example, it can be defined that if the user is brightness sensitive, the screen must be turned into the high contrast mode. It is recommended that the rules be stipulated very clearly (e.g., step by step) and structured as an algorithm in natural language. Depending on the complexity of the UI adaptations, it may be necessary to define several rules. At this stage, there may be no concern with the machine language that will implement the rules. The reference ontology supports the mapping of user characteristics to specific UI elements or adaptation needs. This helps define the adaptation rules. Moreover, existing standards, guidelines and patterns can be considered to help define adequate adaptations (e.g., the W3C Accessibility standard and the Web Content Accessibility Guidelines can be used to help define accessibility adaptations).

(vii) Develop Operational Ontology: consists in producing the operational ontology that will be used at run-time. If there is an operational version of the reference ontology available in the ON, it should be selected. If there is not, the reference ontology must be translated into a machine-readable one. Inference (reasoning) should be carried out to verify the operational ontology consistency. An advantage of using HCI-ON is that it provides the reference ontologies and also their operational versions in OWL. The adaptation rules defined in step (iv) (*Define UI Adaptations*) must be incorporated as axioms into the operational ontology. During run-time, the semantic reasoning engine defined in the system architecture uses the axioms to make inferences on user data and identify the most suitable UI adaptation for a particular user. It is recommended to create some test cases and instantiate them in the operational ontology to verify whether the adaptation rules (axioms) are consistent. For this verification, after the operational ontology population, it is

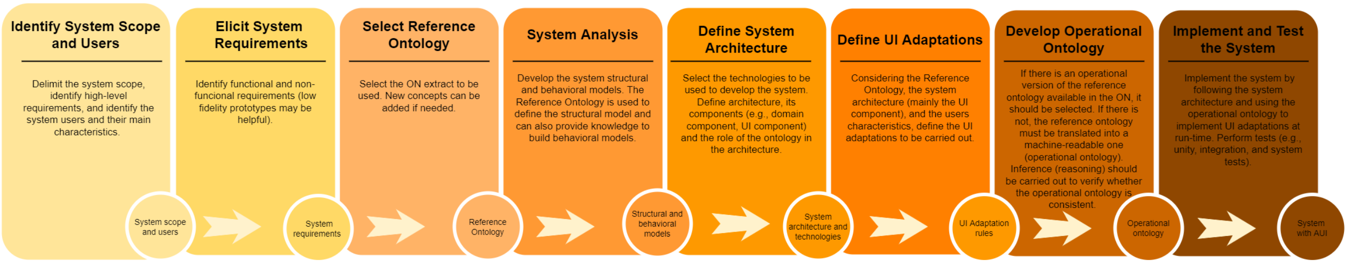

Figure 5: Overview of the ontology-based process to develop AUI systems.

necessary to perform reasoning. This can be done by using tools such as Protégé [7]. The operational ontology (e.g., the OWL file) resulting from this step will be used in the system implementation (next step).

(viii) Implement and Test the System: in this step, the system is implemented by following the system architecture and using the operational ontology to implement UI adaptations at run-time. Unitary, integration and system tests must be carried out to ensure that the system properly meets the established functional and non-functional requirements. Moreover, usability tests are necessary to evaluate the AUI.

It is worth pointing out that although the process is represented as a sequence of steps (for simplification reasons), there is interaction between steps (e.g., it is possible to go back to steps (i) or (ii) after step (iii) to refine user characteristics and requirements based on the reference ontology). Moreover, the process can be performed iteratively.

5 FINAL CONSIDERATIONS

The development of AUI systems is a complex and knowledge-intensive activity. Ontologies have been recognized as important tools for solving knowledge-related problems [13]. Therefore, this paper aimed to explore the use of ontologies, particularly ONs, in the development of AUI systems. Although some works have already used ontologies to develop AUI systems, most of them focus on operational ontologies that are developed specifically for the system where they are used, which hampers reuse and evolution. These works also do not describe the followed process. Moreover, none of them considered the use of ONs. With this research, we give the first step to explore ONs to aid in AUI systems development, leverage the use of ontologies at conceptual and operational levels, and define a systematic process to help in this matter.

The ontology used in the initiative addressed in this paper is an extract of HCI-ON [10, 11], an ON that contains several ontologies addressing HCI subdomains. Thus, other extracts can be considered to extend SNOPI or develop other systems that address different UI adaptations. For example, in HCI-ON, there is the Device Context Ontology, which addresses aspects related to the context of the device where the system runs. Concepts from this ontology can be used to define adaptations considering the device context (e.g., SNOPI could be extended to consider such adaptations). Moreover, as HCI-ON grows, other extracts containing new concepts can be considered to develop more comprehensive and effective AUI systems.

It is important to note that although we have developed SNOPI, it is not the main contribution of this work. The system was developed motivated by some needs of our research group and it is not the end of this research, but a means that helped us explore the use of ontologies from an ON to support the development of AUI systems and give the first step towards an approach with that purpose. This work contributes to the state of the art by exploring an ON to develop AUI systems and evolving a knowledge framework (HCI-ON) to grow knowledge of such systems and AUI. Moreover, the work contributes to practitioners by given the first step to define an ontology-based process that can be used in the development of AUI systems. The work has some limitations that must be considered together with the results presented in this paper. The main limitation is that the ontologies used in the study were developed by the authors and the process resulted from a study also carried out by the authors.

This paper aimed to present the general idea of using an ON to help develop AUI systems and the main results we have obtained so far. This is a long-term research, and there are several involved challenges. For example, regarding the proposed process, it is necessary to refine it, so that other people can properly use it to develop their own AUI systems. It is also necessary to evaluate the process in practical settings and by third parties. Moreover, it should be evolved to be more flexible and cover situations where it is not desirable to use operational ontologies. Concerning the knowledge framework (i.e., the ON), it is necessary to extend the conceptualization in such a way that it describes AUI systems comprehensively (e.g., by considering a wider range of user characteristics, accessibility needs, customizations, context-aware systems, context of use, among others).

Our intention is to work to address challenges like these. Currently, we are dedicating efforts on two main fronts. First, detailing each step of the process to provide the necessary guidelines for other people to use it to develop AUI systems. In this context, our focus is particularly on the steps involving the selection of the ON extract, and its use at both conceptual and operational levels. Second, extending HCI-ON to achieve a comprehensive and consistent knowledge framework to support AUI systems development.

ACKNOWLEDGMENTS

This work is supported in part by CAPES/Brazil (Finance Code 001).

[7] https://protege.stanford.edu/

REFERENCES

[1] Asier Aztiria, Eduardo Castillejo, Aitor Almeida, and Diego López De Ipiña. 2014. Adapting User Interfaces Based on User Preferences and Habits. In *Proceedings of International Conference on Intelligent Environments*. IEEE, Shanghai, China, 9–15. https://doi.org/10.1109/IE.2014.9

[2] David Benyon, Peter Innocent, and Dianne Murray. 1987. System Adaptivity and The Modelling of Stereotypes. In *Human–Computer Interaction–INTERACT '87*, H.-J. Bullinger and B. Shackel (Eds.). North-Holland, Amsterdam, 245–253. https://doi.org/10.1016/B978-0-444-70304-0.50047-9

[3] Fabiano Borges Ruy, Ricardo de Almeida Falbo, Monalessa Perini Barcellos, Simone Dornelas Costa, and Giancarlo Guizzardi. 2016. SEON: A Software Engineering Ontology Network. In *Proceedings of Knowledge Engineering and Knowledge Management*, Eva Blomqvist, Paolo Ciancarini, Francesco Poggi, and Fabio Vitali (Eds.). Springer International Publishing, Cham, 527–542. https://doi.org/10.1007/978-3-319-49004-5_34

[4] Amani Braham, Félix Buendía, Maha Khemaja, and Faiez Gargouri. 2021. User interface design patterns and ontology models for adaptive mobile applications. *Personal and Ubiquitous Computing* (jan 2021), 1–17. https://doi.org/10.1007/s00779-020-01481-5

[5] Suárez-Figueroa Mari Carmen, Gómez-Pérez Asunción, Motta Enrico, and Gangemi Aldo. 2012. *Introduction: Ontology Engineering in a Networked World*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–6. https://doi.org/10.1007/978-3-642-24794-1_1

[6] Roberto Carraretto and Joao Paulo A. Almeida. 2012. Separating Ontological and Informational Concerns: Towards a Two-Level Model-Driven Approach. In *Proceedings of IEEE 16th International Enterprise Distributed Object Computing Conference Workshops*. IEEE, Beijing, China, 29–37. https://doi.org/10.1109/EDOCW.2012.14

[7] John Carroll. 1997. Human-computer interaction: Psychology as a science of design. *Annual review of psychology* 48 (02 1997), 61–83. https://doi.org/10.1146/annurev.psych.48.1.61

[8] Alan Cooper. 1999. *The Inmates are Running the Asylum*. Vieweg+Teubner Verlag, Wiesbaden, 17. https://doi.org/10.1007/978-3-322-99786-9_1

[9] Simone Dornelas Costa, Monalessa Perini Barcellos, and Ricardo de Almeida Falbo. 2021. Ontologies in human–computer interaction: A systematic literature review. *Applied Ontology* 16, 4 (nov 2021), 421–452. https://doi.org/10.3233/AO-210255

[10] Simone Dornelas Costa, Monalessa Perini Barcellos, Ricardo de Almeida Falbo, and Murillo Vasconcelos Henriques Bittencourt Castro. 2020. Towards an Ontology Network on Human-Computer Interaction. In *Proceedings of the 39th International Conference on Conceptual Modeling*, Gillian Dobbie, Ulrich Frank, Gerti Kappel, Stephen W Liddle, and Heinrich C Mayr (Eds.). Springer International Publishing, Cham, 331–341. https://doi.org/10.1007/978-3-030-62522-1_24

[11] Simone Dornelas Costa, Monalessa Perini Barcellos, Ricardo de Almeida Falbo, Tayana Conte, and Káthia M. de Oliveira. 2022. A core ontology on the Human–Computer Interaction phenomenon. *Data & Knowledge Engineering* 138 (mar 2022), 101977. https://doi.org/10.1016/j.datak.2021.101977

[12] Ricardo José de Araújo, Julio Cesar Dos Reis, and Rodrigo Bonacin. 2016. Ontology-Based Adaptive Interfaces for Colorblind Users. In *Proceedings of Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices*, Margherita Antona and Constantine Stephanidis (Eds.). Springer International Publishing, Cham, 27–37. https://doi.org/10.1007/978-3-319-40250-5_3

[13] Christina Feilmayr and Wolfram Wöß. 2016. An analysis of ontologies and their success factors for application to business. *Data & Knowledge Engineering* 101 (jan 2016), 1–23. https://doi.org/10.1016/j.datak.2015.11.003

[14] Dave Gray, Sunni Brown, and James Macanufo. 2010. *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers* (1st ed.). O'Reilly Media, Inc. 266 pages.

[15] Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What Is an Ontology? In *Handbook on Ontologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–17. https://doi.org/10.1007/978-3-540-92673-3_0

[16] Giancarlo Guizzardi. 2007. On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In *Proceedings of the Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*, Olegas Vasilecas, Johan Edler, and Albertas Caplinskas (Eds.). IOS Press, Amsterdam, The Netherlands, The Netherlands, 18–39. http://dl.acm.org/citation.cfm?id=1565421.1565425

[17] Anthony Jameson. 2002. *Adaptive Interfaces and Agents*. L. Erlbaum Associates Inc., USA, 305–330.

[18] Talia Lavie and Joachim Meyer. 2010. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies* 68, 8 (2010), 508–524. https://doi.org/10.1016/j.ijhcs.2010.01.004

[19] Oppermann R. 1994. *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. CRC Press, Boca Raton, Flórida, EUA.

[20] Nilanka Rathnayake, Dulani Meedeniya, Indika Perera, and Anuradha Welivita. 2019. A Framework for Adaptive User Interface Generation based on User Behavioural Patterns. In *Proceedings of 2019 Moratuwa Engineering Research Conference (MERCon)*. IEEE, Moratuwa, Sri Lanka, 698–703. https://doi.org/10.1109/MERCon.2019.8818825

[21] Jiri Sebek, Michal Trnka, and Tomas Cerny. 2015. On Aspect-Oriented Programming in Adaptive User Interfaces. In *Proceedings of 2015 2nd International Conference on Information Science and Security (ICISS)*. IEEE, Seoul, Korea (South), 1–5. https://doi.org/10.1109/ICISSEC.2015.7371024

[22] Stefan Seedorf, Fzi Forschungszentrum Informatik, and Universität Mannheim. 2006. Applications of Ontologies in Software Engineering. In *Proceedings of Workshop on Semantic Web Enabled Software Engineering (SWESE) on the ISWC*. Citeseer, 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006), held at the 5th International Semantic Web Conference (ISWC 2006), Athens, Geórgia, 5–9.

[23] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering* 25, 1-2 (mar 1998), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

[24] Enes Yigitbas, Klementina Josifovska, Ivan Jovanovikj, Ferhat Kalinci, Anthony Anjorin, and Gregor Engels. 2019. Component-Based Development of Adaptive User Interfaces. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Valencia, Spain) *(EICS '19)*. Association for Computing Machinery, New York, NY, USA, Article 13, 7 pages. https://doi.org/10.1145/3319499.3328229