Ontology-driven Elicitation of Multimodal User Interface Design Recommendations

Tom Tourwé [a] Elena Tsiporkova [a] Nicolás González-Deleito [a]
Anna Hristoskova [b]

[a] *ICT & Software Engineering Group, Sirris*
Diamant Building, Reyerslaan 80, B-1030 Brussels, Belgium
[b] *Ghent University, Department of Information Technology - IBBT*
Gaston Crommenlaan 8 (Bus 201), B-9050 Ghent, Belgium

Abstract

In this paper, we argue that current design guidelines for multimodal user interface design focus mostly on high-level design objectives and do not capture and reflect on the considerable practical experience and valuable expert knowledge that interface designers rely on. We propose an ontology-driven modelling framework, which allows to capture the domain and expert knowledge available within the interface design community and to support designers in their daily design tasks by eliciting user and application dependent design recommendations. We illustrate how this framework can be used in practice with a concrete case study devoted to multimodal interface design for the purpose of emergency response applications.

1 Introduction

Applications utilising multimodality for user input and information presentation are no longer novelty nowadays. Combination of touch, speech and other modalities, such as lip movements and gaze, have been exploited in the recent years thanks to advanced input/output technologies (e.g. see Oviatt [6]). Development of multimodal applications is inherently complex due to the fact that these applications are usually targeting complex data-rich environments and need to address challenges as data overload, requirements for improved recognition performance, support for time and attention sharing, etc. [9]. Moreover, in order to use the best suitable modality at a given time, the application must also be context-aware. The challenge is to design multimodal interfaces which can reliably interpret continuous input from different visual, auditory, and other sources in order to make an accurate context assessment and response planning in support to the user's tasks.

Parallel with the evolution towards multimodal applications, the research toward the establishment of formal principles and guidelines for multimodal interaction design is gaining increasing interest and importance in recent years (e.g. [7, 8, 2]). However, as observed by Sarter [9], the existing guidelines mostly focus on high-level design objectives and do not provide support on how to map them to the needs of an actual application. They do not capture and reflect on the considerable practical experience and valuable expert knowledge that interface designers rely on during their daily activities. Moreover, a considerable gap exists between the theory (formal guidelines) and the practice of multimodal human interface design, as different experts might approach the same interface design tasks in different ways based on personal expertise, background and intuition.

Our aim in this article is to work toward bridging this gap via the application of semantic technologies (e.g. ontologies) for capturing the available domain and expert knowledge in the field of multimodal interface design. There are several advantages associated with such an approach: it guarantees a uniform approach across different designers within the same organisation, allows for semantic inter-usability of the formal guidelines across different applications and domains, facilitates context representation, and is open to allow for knowledge evolution and growth. In this context, the semantic framework proposed in this article is well aligned with the proposition of Woods et al. [11] to consider guidelines as "a synthesis or abstraction of current knowledge ... and a stimulus to the growth of knowledge".

The next section discusses current practices in multimodal interface design and provides the rationale for our research. Section 3 introduces our current semantic modelling framework implementation in Protégé 4 [4]. The ultimate goal is to provide formal means for capturing and modelling the domain knowledge and best practices available within the designer community and subsequently enable reasoning about this knowledge in order to offer support to the design practitioners by automatically deriving and recommending selection and combination of appropriate modalities optimally mapped to the users' tasks, environment and the type of information. Section 5 illustrates how this semantic framework can be used in practice with a concrete case study devoted to multimodal interface design for the purpose of emergency response applications.

2 Multimodal User Interface Design Guidelines

An exhaustive survey on the current state-of-the-art of multimodal interfaces was published recently [2]. It covers the foundations and features of multimodal interaction, current developments in modelling languages and programming frameworks, and existing principles and guidelines for multimodal interface design. Several authors worked on establishing formal principles for multimodal user interface design. Reeves et al. [8] define a set of principles divided in six different categories of guidelines: requirements specifications, designing multimodal input and output, adaptability, consistency, feedback and error preventions/handling. Some of the included principles are: design for the broadest range of users and contexts of use, address privacy and security, maximise human cognitive and physical abilities, integrate modalities in a manner compatible with user preference, context, and system functionality, etc. Although these principles represent a valuable methodological advancement in the domain of multimodal interaction design, they are of a little practical use to the daily activities of the designers.

Sarter [9] reviewed the existing design guidelines for multimodal information presentation, approaching the problem from the point of view of main decisions and considerations involved in the multimodal interface design and thus identified four themes of guidelines:

- *selection* of modalities;
- *mapping* modalities to tasks and types of information;
- *combination*, synchronisation and integration of modalities;
- *adaptation* of multimodal information input and presentation to accommodate changing task contexts and circumstances.

Sarter's study covers and discusses in detail a broad range of existing guidelines, recommendations, best practises etc. associated with each of these themes. The common shortcomings of these guidelines are also well addressed, e.g. most guidelines do not reflect adequately the human perception specifics, do not include justification for the made recommendations, are not specific to multimodality, lack information about how to move from guidelines to a concrete implementation.

Sarter considers that most of the above shortcomings are mostly due to the fact that there remain a substantial number of open research problems in the area of multimodal information processing. We agree partially with this statement. However, we consider that future research advancements would not be sufficient to address all these shortcomings since the guidelines resulting from research would always remain of a rather conceptual and less empirical nature and thus of little *practical* use. In our opinion, an extensive intuitive and empirical knowledge base exists already within the designer community and the challenge is to develop methods, formal languages and frameworks that allow for capturing and exploiting this knowledge. The semantic modelling framework proposed in this work is an initial attempt in this direction. The selected approach is based on Sarter's insight of considering the interface design process from the perspective of the decision dilemmas designers face daily when executing their design tasks.

3 Ontology-driven Elicitation of Recommendations

We propose here a semantic modelling framework (see [10]), which allows to capture general *domain knowledge* and *expert knowledge*. The former considers all factual information relevant to the Human-Computer

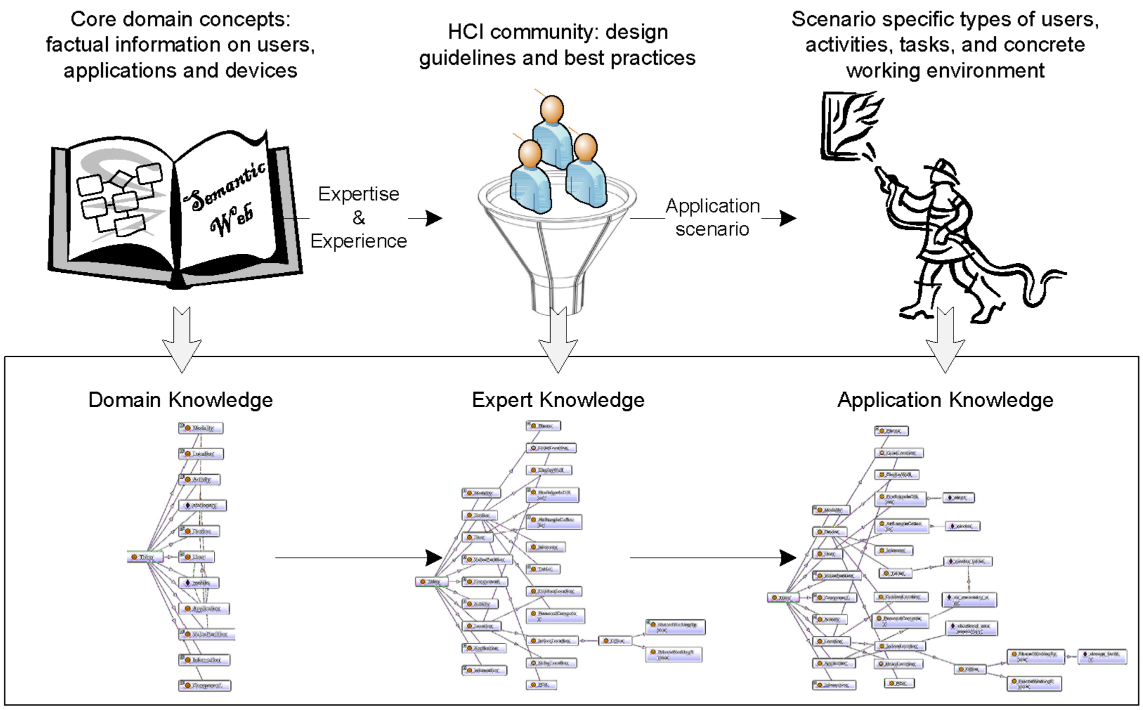

Figure 1: Schematic overview of the different levels of semantic modelling abstraction: domain, expert and application knowledge

Interaction (HCI) domain, while the latter attempts to capture the available and well established in the HCI community guidelines and best practices related to multimodal application design. Both domain and expert knowledge are described via an ontology, a formal representation of knowledge by a set of key domain concepts and the relationships between those concepts. We complement this with *application-specific knowledge* and illustrate how the framework supports the decision-making of which modalities are suitable candidates for an application.

The proposed method structures the semantic modelling in three levels of abstraction as presented in Figure 1. The first two levels, domain and expert knowledge, are modelled within our "core" HCI ontology, while the application-specific knowledge is defined in an additional application-specific ontology, which is an extension and instantiation of the core ontology and reflects the concrete context of use of the application.

4 Core HCI ontology

4.1 Competency Questions

To determine the scope of an ontology and identify the questions that an ontology-based knowledge repository should be able to answer, we need to state so-called *competency questions* [3]. Subsequently, from these questions the information that needs to be contained within the ontology, i.e. the concepts and the relationships between concepts, can be derived.

We consider that at this stage of the research, our ontology should be able to answer the following competency questions, which are inspired by the major themes identified by Sarter [9] (see above):

- Which input and output modalities are available to the user of an application?
- What are the different factors that affect the use of particular input and output modalities?
- What are the appropriate (combinations of) modalities to support users in a particular task?

Based on this list of questions, the ontology will include information on users, their context and tasks, the type of input and output information, the devices they use and their specific capabilities, and usage characteristics and constraints of particular user-interface modalities. The next section discusses the key ontology domain concepts and relationships.

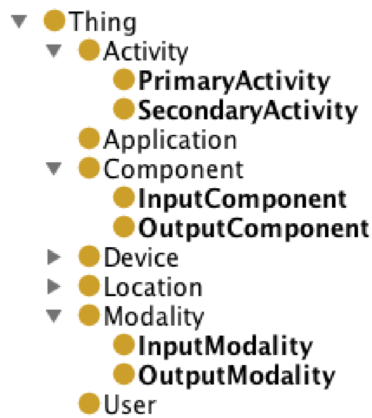

Figure 2: High-level domain concepts in the semantic modelling framework

Name	**Specifies ...**	**Properties**
is_located_in	a user is located in a location	functional, inverse property: **contains**
uses	a user uses an application	inverse property: **used_by**
performs	a user performs an activity	
runs_on	an application runs on a device	
has_component	a device features a component	
is_near	a user is near to another user	
has_noise_level	the noise level of a location	functional
has_access_to	an application has access to a component	defined as property chain: **runs_on** and **has_component**
used_in	an application is used in a location	defined as property chain: **used_by** and **located_in**
requires	an activity requires a capability	
supports_activity	an application supports performing an activity	
supports_modality	a component supports a modality	
has_property	a user has a characteristic	

Table 1: Relationships between high-level domain concepts

4.2 Key Domain Concepts

We define the core HCI ontology, consisting of general and high-level key domain concepts as depicted in Figure 2. The class **User** represents a user of an application, which itself is represented by the class **Application**. The class **Location** represents the physical location where the user is performing his activities and using the application. Locations are considered as being an **IndoorLocation** or an **OutdoorLocation**. The class **Device** represents the device that the user is using and on which applications run, while the class **Component** represents the different components of a device. The latter class is further specified as being either an **InputComponent** (e.g. a microphone) or an **OutputComponent** (e.g. a speaker). Different components support a different **Modality** (e.g. a microphone supports voice input). The class **Activity** represents the activities that a user can engage in, subdivided into **PrimaryActivity** and **SecondaryActivity**.

These concepts are related through the relationships described in Table 1. Some relationships are defined through *property chains* which enable them to be automatically inferred from other relationships. For example, if we know that an application A is used by a specific user U, and that this user is located in a particular location L, we can infer that application A is used in location L. More formally

$$A \textbf{ used\_by } U \wedge U \textbf{ located\_in } L \rightarrow A \textbf{ used\_in } L.$$

4.3 Modelling Domain Knowledge

We model domain knowledge by specifying necessary conditions for the key domain concepts in our core ontology. We consider domain knowledge to be any factual information about users, applications and devices that potentially influences the decision about which modality to provide. This includes obvious information such as the specific input/output modality supported by a component of a device, but also information such as physical and social aspects of the user's working environment, or particular aspects of the nature of the activity (e.g. primary and secondary tasks).

For example, a contemporary computer features a microphone and speakers, which we model by defining two necessary conditions on a class **PersonalComputer**, a subclass of class **Device**, as follows:

has_component value *microphone*
has_component value *speakers*

where *microphone* is an instance of class **InputComponent**, and where *speakers* is an instance of **OutputComponent**. Modelling the class **PersonalComputer** in this way, we formally define that any personal computer in our domain necessarily includes both a microphone and speakers, and hence necessarily supports voice input and audio output modalities.

As an example of information regarding the user's environment, we define a class **PrivateWorkingSpace** as an **IndoorLocation** that is quiet by adding a condition **has_noise_level some Quiet**. Similarly, a **SharedWorkingSpace** is an **IndoorLocation** that is loud, because of the presence of different people in the environment:

has_noise_level value *loud*

4.4 Modelling Expert Knowledge and Eliciting Recommendations

As already mentioned above, expert knowledge is understood as a set of design guidelines which capture the expertise and experience of the HCI practitioners. They describe applicability conditions and constraints for the use of a particular multimodal interface.

We capture design guidelines via the Semantic Web Rule Language (SWRL) [5], which is used for coding procedural knowledge description in ontologies in the form of rules. This allows existing description logic reasoners such as Pellet [1] to execute data transformations defined in SWRL rules. Below, we illustrate how our framework supports several of such guidelines.

The main idea is to have the reasoner derive extra properties that hold for an application, **could_use_modality** and **cannot_use_modality**, describing whether or not an application can use a particular modality.

For instance, having in mind that the accuracy of voice technology is heavily dependent on environmental noise conditions (e.g. background noise), we can rule out interaction with an application through vocal commands and audio output if the user is using the application in a noisy environment. This can be expressed as follows in SWRL:

Application(?application), NoisyLocation(?location), used_in(?application, ?location) →
cannot_use_modality(?application, audio_output),
cannot_use_modality(?application, voice_input)

In the same fashion, interaction through a touch interface can be recommended when the following conditions and constraints are met:

- the application is used by a user who is mobile i.e. needs to move around to perform his primary activity;
- the primary activity of the user does not require the use of both hands or in other words the user has at least one hand free to control the application.

The above constraints can be modelled as follows:

Application(?application), PrimaryActivity(?activity), User(?user),
has_property(?user, mobile), requires(?activity, no_hands),
uses(?user, ?application) → could_use_modality(?application, haptic_input)

Application- and domain-specific rules can be defined in application- and domain-specific sub-ontologies of our core ontology, as illustrated in the next section.

5 Case Study: Multimodal Interface Design for Emergency Response Application

5.1 Context of Use

In order to illustrate how the semantic model defined in the previous section can be used to elicit modality design recommendations, we consider two concrete application scenarios derived from the emergency management demonstrator of the ASTUTE project[1], in which the authors from Sirris participate. This demonstrator considers a fire in an industrial site, and involves the coordination of all the relevant stakeholders in order to evacuate the site, extinguish the fire and bring the area affected by the fire back to a usable state.

In this context, our original core ontology needs to be complemented and extended by creating an ontology with relevant application-specific knowledge. For instance, the different types of users involved in this scenario (fire fighters, fire commanders, fire station dispatchers, air sampling collectors, emergency communication managers, medical experts, company employees, etc.), their activities and tasks (fire fighting, locating water supplies, rescuing company employees that could not leave a building, logging relevant information, defining security perimeters in the presence of dangerous substances, etc.), and the concrete working environment they are located in (an administrative office where the fire started, a storage facility with smoke and high temperatures, outside a building where dangerous substances might be being spread in the air, inside a medicalised tent, etc.).

Our two concrete scenarios involve two rather different types of stakeholders, in terms of role, context and needs:

- *an air sampling collection team* that needs adequate support to perform optimally their activities in the field around the fire location;
- *a fire brigade officer* who is coordinating the firemen fighting the fire emergency and communicating with the dispatching control room.

5.2 Application-specific Ontology

The air sampling collection team frequently measures the quality of the air, its speed and direction, as well as other weather conditions at different locations around the industrial site in order to evaluate how dangerous substances are actually being spread. Members of this team keep a record of the measurements in an application running on a **mobile device**. Due to regulations, they are required to wear gloves and a mask while performing the measurements. Finally, measurements take place at locations sufficiently far away from the location of the fire so that the working environment of the members of this team can be considered most of the time as quiet. We model this application-specific knowledge by:

- defining an instance of **PrimaryActivity** called *measuring_air_quality*, which **requires** *both_hands*;
- defining an instance of **Application** called *air_measuring_app* which **runs_on** *nicolas_tablet*;
- defining an instance of **OutdoorLocation** called *industrial_site*, which **has_noise_level** value *quiet*;
- defining a subclass of **User** called **AirSampleCollector** with necessary conditions stating that each instance **performs** the *measuring_air_quality* activity, **uses** the *air_measuring_app*, and **is_located_in** an *industrial_site*;
- defining an instance of **AirSampleCollector** called *nicolas*;
- defining an instance of the **Tablet** called *nicolas_tablet*;
- stating that the *air_measuring_app* **runs_on** *nicolas_tablet*;

[1]ASTUTE is a large EU project (www.astute-project.eu) which aims at defining a reference architecture for the development of human machine interactions, targeting proactive information retrieval and delivery based on the situational context, as well influenced by information content and services, and user state information. The ultimate goal is to design intelligent multi-modal interfaces enabling to determine which information and services to push to the user at the right time via the appropriate modality. The approach will be verified in several different industrial demonstrators in the domain of avionics, automotive and emergency management.

The semantic engine can now combine this knowledge with the domain-specific knowledge and the expert knowledge to automatically suggest that voice could be used as input modality. It does so by deriving that the relationship **could_use_modality** holds between the *air_measuring_app* application and the *voice_input* modality.

The air sampling collector also uses the air sampling record application while back in his office to perform some statistical analysis on the data and produce formal report. In such circumstances, the air sampling collector will certainly choose to use traditional interface modalities like keyboard and mouse. This can be accordingly coded in the ontology.

In our second scenario, we consider a fire brigade officer situated at the emergency site. The fire brigade officer is coordinating the fire men fighting the fire and communicating with the dispatching control room. He is moving around the site, carrying a mobile device that is running an application supporting situational awareness, allowing him to be aware of what is happening and helping him decide what is the appropriate course of action. Understandably, the emergency site is quite noisy, as people deploy heavy materials, shout instructions to each other, find themselves in a stressful situation, etc. It is thus logical that vocal and audio technologies are excluded as potential interface modalities for the situational awareness application. We model this application-specific knowledge as by

- defining an instance of **PrimaryActivity** called *coordinating_fire_brigade*, which **requires** *no_hands*;
- defining an instance of the class **Application** called *situational_awareness_app* which **supports** *coordinating_fire_brigade*;
- defining an instance of **Location** called *emergency_site*, which **has_noise_level** value **loud**;
- defining **FireBrigadeOfficer** as a **User** who **has_property** *mobile*, who **performs** the *coordinating_fire_brigade* activity, who **uses** the *situational_awareness_app* application, and who **is_located_in** the *emergency_site*;
- defining an instance of **FireBrigadeOfficer** called *elena*;

With this additional application knowledge, the semantic engine can automatically derive that haptic input (i.e. touch) could be considered as a modality, by deriving that the **could_use_modality** holds between the *situational_awareness_app* application and the *haptic_input* modality. In addition, the engine derives that audio output modality cannot be used, due to the fact that the officer is working in a noisy environment, and that the manual input (e.g. by means of a keyboard) cannot be used, because the officer needs to be mobile.

6 Conclusions and Future Work

This paper presents an initial attempt to formally model and exploit relevant HCI domain knowledge and practitioners' expertise in support of selecting appropriate modalities during the human machine interface design process. The work presented here is inspired and results from our interactions with HCI practitioners.

The framework will be validated in different application domains of the ASTUTE project, by incorporating the domain- and application-specific knowledge of the different demonstrators (avionics, automotive, emergency dispatching).

Besides further refinement of the presented semantic modelling framework (e.g. incorporating standard ontologies for devices and locations), future research includes considering other competency questions that our ontology could support (such as *what are the important functionalities/tasks?* and *what kind of devices could an application run on?*), expanding toward broad range of working contexts and types of users, and supporting synchronisation of modalities in time. All this research will be performed in close collaboration with HCI practitioners.

7 Acknowledgements

This research is supported by the ARTEMIS JU and IWT (www.iwt.be) through the ARTEMIS project (www.artemis-project.eu), and by Innoviris (www.innoviris.be) through the VariBru project (www.varibru.be).

References

[1] Pellet: OWL 2 reasoner for Java.

[2] B. Dumas, D. Lalanne, and S. Oviatt. Multimodal interfaces: A survey of principles, models and frameworks. *Human Machine Interaction*, pages 3–26, 2009.

[3] M. Gruninger, M.S. Fox, et al. Methodology for the Design and Evaluation of Ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI*, volume 95, 1995.

[4] Matthew Horridge. *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. The University Of Manchester, 1.2 edition, March 2009.

[5] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML., 2004.

[6] S. Oviatt. Advances in robust multimodal interface design. *Computer Graphics and Applications, IEEE*, 23(5):62–68, 2003.

[7] S.L. Oviatt. Multimodal interfaces. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, pages 286–304, 2003.

[8] L.M. Reeves, J. Lai, J.A. Larson, S. Oviatt, TS Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.C. Martin, et al. Guidelines for multimodal user interface design. *Communications of the ACM*, 47(1):57–59, 2004.

[9] N.B. Sarter. Multimodal information presentation: Design guidance and research challenges. *International journal of industrial ergonomics*, 36(5):439–445, 2006.

[10] E. Tsiporkova, T. Tourwé, and N. González-Deleito. Towards a semantic modelling framework in support of multimodal interface design. In *Proceedings of the 13th IFIP TC13 Conference on Human-Computer Interaction, INTERACT 2011 (to appear)*, 2011.

[11] D.D. Woods, L. Johannesen, and S.S. Potter. The sophistry of guidelines: Revisiting recipes for color use in human-computer interface design. In *Proceedings of the Human Factors and Ergonomics Society 36th Annual Meeting*, pages 418–422. Human Factors and Ergonomics Society, 1992.