Journal of Universal Computer Science, vol. 17, no. 7 (2011), 1078-1088
submitted: 30/10/10, accepted: 15/3/11, appeared: 1/4/11 © J.UCS

Ontology-based User Interface Development: User Experience Elements Pattern

Syed K. Shahzad
(Graz University of Technology, Graz, Austria
shahzad@student.tugraz.edu)

Abstract: The user experience of any software or website consists of elements from the conceptual to the concrete level. These elements of user experience assist in the design and development of user interfaces. On the other hand, ontologies provide a framework for computable representation of user interface elements and underlying data. This paper discusses strategies of introducing ontologies at different user interface layers adapted from user experience elements. These layers range from abstract levels (e.g. User needs/Application Objectives) to concrete levels (e.g. Application User Interface) in terms of data representation. The proposed ontological framework enables device independent, semi-automated GUI construction which we will demonstrate at a personal information management example.

Keywords: GUI, Personal Information Management, User Interface Ontology, HCI, Knowledge Management
Categories: H.1.2, H.5.2, H.5.3, I.2.4, K.6

1 Introduction

Guarino, Smith and Gruber introduced the formalization of abstract domain concepts by using ontologies [Guarino 95] [Smith 04] [Gruber 93]. By combining ontologies with logical theory, machines can be enabled to compute formalized conceptual models using shareable, domain specific ontologies. Moreover, formal ontologies ensure the meaning and consistency of the presented concepts and allow to unambiguously share domain specific concepts across decentralized information systems.

Ontologies have been used in information systems on several levels, like for example for integrating databases, as business logic or for constructing Graphical User Interfaces (GUI). Additionally, ontologies have been exploited in software development processes. Uschold for example combined formal ontologies with model driven software development [Uschold 08]. Uschold outlined that ontological model driven software development improves the development of complex software systems. Ontology based modelling carries the domain concept to the entire software development process.

In our work we present an approach for mapping formal ontologies to GUI automatically, so that the GUI preserves the domain specific properties of the underlying domain ontology. Such a mapping supports device independent GUI construction as well as semi automatic GUI modelling. We discuss our approach along a Personal Information Management (PIM) example. We used vCard/hCard as

a standard ontological model for PIM. We used Jena and SWT for mapping vCard ontologies to GUI.

However, the use of ontologies within the GUI development process also provides technology independent modelling. Moreover, it allows to add context-aware properties at the development level depending on the targeted devices and user roles.

The next section discusses elements of user experience providing the layers of user's perception. In section 3, we outline the properties of the formal ontology which should be presented at the GUI level and conclude our work by combining User Interface Ontology (UIO) and Personal Information Management.

2 User Experience Elements

Garrett [Garrett 02] introduces five elements of user experience by concepts underlying software or a website [see Figure 1]. These application concepts summarize the goals a software system should pursue. Garrets elements collectively introduce different levels of such application concepts represented in an information system and will be described in the following.

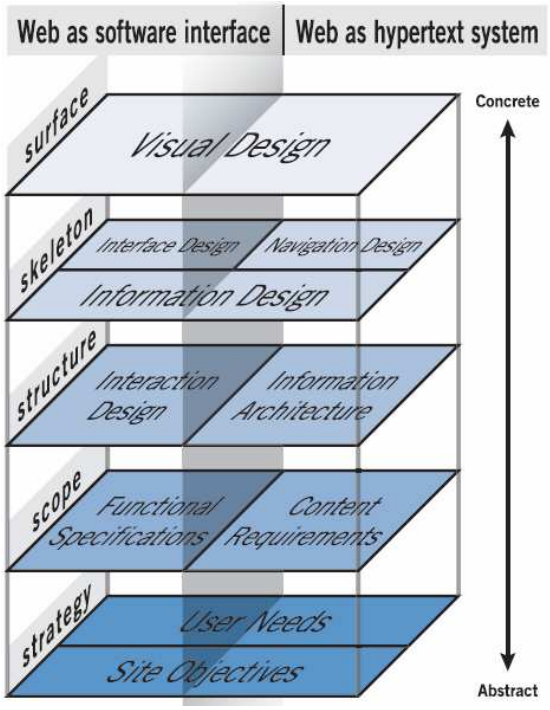

Figure 1: Adapted User Experience Elements by Garrett [Garrett 02]

By referring to Garrett's model, we argue that representation and direct mapping of ontology from the base layer (i.e. strategy) to the presentation layer (i.e. surface)

can help users to perceive the targeted domain concept. These elements can also be considered as layers of UI design and development. We adapted Garret's layered approach to GUI development to follow through the elements of user perception. The ontological framework provides the capability for a formal specification of the concepts. It also enables the designer and developer of the website to implement ontologically specified domain concepts. Following user psychology also enables the designer to make a user centred design. Separating the different roles in GUI construction on the different levels of user experience reduces development time and allows for independent optimization of each level.

Level of User Experience	Ontologcial Implementation	Example Application using Ontological Framework
Surface	Graphics Look and Feel	UIO Implementation at Graphical Library SWT, OpenGL, GTK, wxWidgets, QT
Skeleton	User Interface Ontology	Customized Textbox, List box, Selection Box, Date/Time tool, Containers, Buttons
Structure	Domain Ontology	vCard, hCard
Scope	Vocabulary (for Entities and Relations) Relations also represent Functions	Name, Address, Date of Birth, Email, Phone Number, Family Name, Zip Code
Strategy		Personal Information Management

Table 1: Adapted User Experience Elements by Garrett [Garrett 02]

Ontologies can be introduced to each layer for integrating the different user experience levels as follows [see also Table 1]:

1. Garret represented the strategy as an abstract concept which the user abstractly defines. It specifies the user needs and objectives of the targeted GUI. At this level, there is no explicit ontological formulation in order to represent this abstract strategy.
2. Scope talks about what exists within the boundary of the domain. In this work, it refers to the details of the concepts and sub-concepts within the domain. It merely addresses the vocabulary used, but not its structure, which is part of the next level.

3. Structure defines the logical structure of the user interface. Domain ontologies specify this via relationships among concepts and sub-concepts forming the logical structure of the user interface. It provides a hierarchical structure of whole and part, but also associated concepts. This ontological structure makes a map of information arrangement and navigation in related concepts.
4. The skeleton level provides the representation and interaction methods of an user interface. Ontologically formalized structures represented as User Interface Ontology allows specification of such a skeleton and translates the logical domain structure of step 3 into a user interface layout which is rendered in the next step.
5. Surface is the concrete implementation of the skeleton. It is the User Interface Ontology implemented through any computer graphical library.

In the following example of mapping vCard onto GUIs via those 5 steps, we have dealt with visualization of vCard to the GUI only. Interaction methods, update functions and initiation with UI events will be discussed in future research through mapping functional ontology to User Events.

2.1 Conceptualization through Ontological Modelling

Gruber [Gruber 93] suggested the properties of formal ontologies for conceptualization. Uschold introduced ontological modelling for model driven software development to use the properties of conceptualization. We use formal ontologies as a base structure for GUI development.

Formal ontology specifies the structure. UI Ontology will use the vocabulary of each entity concept and sub-concept, specified by formal ontology. These concepts and sub-concepts can be carried to the structured layer using ontological relations. Automatic mapping functions from formal ontology to GUI can maintain properties of conceptualization at GUI.

2.2 Personal Information Management (vCard/hCard)

Internet Mail Consortium (IMC)[1] has defined a standard for Personal Data Interchange as vCard (RFC2426). This standard has been broadly implemented (e.g. Apple's "Address Book", Microsoft Outlook) ensuring interoperability. This paper discusses factors and issues in mapping vCard to GUI. vCard provide an ontological structure for personal information storage and representation and hence we use vCard as example ontology for personal Information. The Internet Mail Consortium states, that vCard can be used to forward personal data to an electronic mail message. While integrating vCard support into an application, an implementer must consider a number of UI implications. Most applications provide some levels of support for interacting with other applications. RFC2426 mentioned three ways to use vCard the File System, the Clipboard, and Drag/Drop techniques. It also provides two kinds of grouping: first grouping multiple vCard and second grouping related properties within the same vCard. The IMC White paper also argues that full potential of the vCard

[1] http://www.imc.org/pdi/

technology can be better utilized by an application that supports the vCard specification in each of these UI forms.

Identification Properties	Name	Given
		Family
Address Properties	Street Address	
	City	
	Country	
	Zip Code	
Telephone Number	Country Code	
	Network Code	
	Phone Number	
....		

Table 2: Structure of vCard properties

The grouping of vCard elements provides an ontological structure [see also Table 2] forming a domain ontology. As outlined above, such domain ontology can be used at the structure level to form the logical structure of the GUI. There are also some implicit relations within vCards realized as different groups or sets of attributes.

vCard properties can be grouped and prioritised according to the context and relationships. The values of grouped properties can be represented as one value but with different joining structures and delimiters.

3 Ontological Framework for User Interface Development

The ontological framework includes three major sections (i.e. vocabulary, Domain Ontology and User Interface Ontology). These sections are the layers of development procedure.

3.1 Vocabulary (Scope)

The section Vocabulary contains the concepts and sub-concepts that exist in the domain without stating relationships. It is list of attributes of the domain ontology. vCard standard (RFC2426) provides the list of attributes being used for PIM.

3.2 Domain Ontology (Structure)

The section Domain Ontology specifies the relationship of the attributes. It provides a structure for arranging attributes at GUI. In our example vCard provides a taxonomy of information [see Table 2]. It provides the way personal information should be arranged and grouped, e.g. first name and last name should be together, city and country information will be in address group. Though it is most of the time arranged

in the same way, it is not a rule by the base model in designing and development of the user interface.

This structure defines the taxonomy as a rule which cannot be avoided in GUI development.

3.3 User Interface Ontology (Skeleton)

Uschold encourages automated code generation for UI from ontological models [Uschold 08]. But most of the research talks and presents automation and translation of ontologies to GUI [Alexander, 03] [Furtado 02] [Liu 05]. Paulheim provides a method to implement ontologies to UI through plug-ins [Paulheim 09].

We used the User Interface Ontology (UIO) as one part of the core development process and a base for designing UI, rather than introducing ontologies separately or in parallel to the main design procedure. User Interface Ontology (UIO) describes concepts and relationship of GUI objects and interaction methodology. UIO considers that the domain ontology needs to be interfaced to the user. It specifies the representation of each entity in the vocabulary depending on the type of entity, the relationship specified in the domain ontology and the role of the user. UIO considers the role of users to generate GUI according to the context.

UIO contains domain ontology user interface concepts and their properties at the user interface level. Domain ontology defines a vocabulary and a structure. At UIO level, vocabulary is associated with conceptual visualization according to its type (textual, image, multi-media, map or a group) and Structure provides arrangements of the visualization at User Interface.

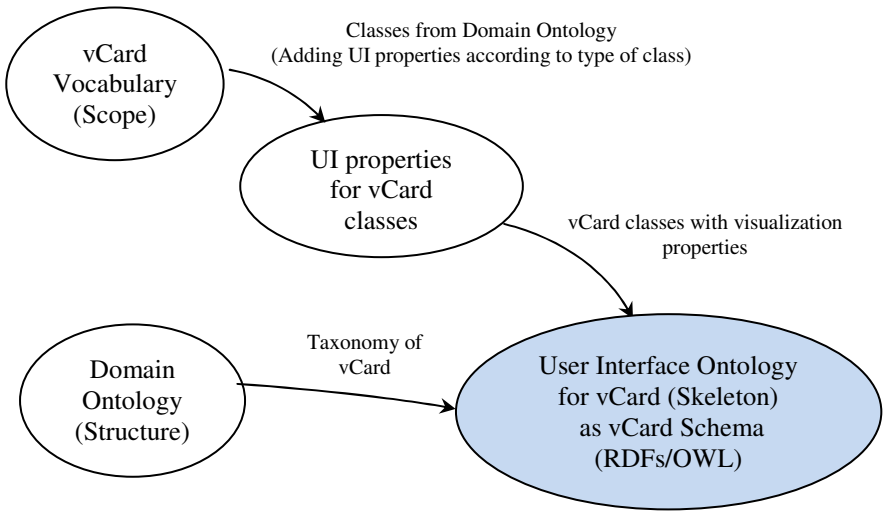

Figure 2: User Interface Ontology for vCard (Skeleton)

User Interface properties were added to the Domain Ontology classes carried out in the ontology. These properties specify the visualization mechanism based on the

class and type of the instance. Literals, groups, sets and lists can be presented accordingly [see Figure 2].

In our example of PIM, vocabulary and taxonomy can be read from vCard provided as RDFS. In vCard, Name is a group of two items, "Given and Family". These sub-parts of names are literals (textual data). So visualization properties can be associated at this small part and class and sub-class. Additional UI properties based on context, like font, colour or size, can also be added later on at skeleton level via the UIO.

UI is two-way communication "from and to the user". The concepts discussed in UIO are the presentation of a domain concept, sub-concept and the relation to the user as to the user communication; it also discusses the interaction methodology and user actions and ontological responses. Functional Ontology is part of future work and conceptual user actions to concrete event handling will also be part of future work.

3.4 User Interface development (surface)

This section discusses UI development methodology through UIO.

3.4.1 Customized User Interface Control

GUI controls can be customised and associated with the domain concepts which are going to be represented at GUI. So, text box, list box, combo box, radio button or even a frame are associated with a domain concept presented in vocabulary according to their UI properties. IBM Semantic Layered Research Platform[2] has also worked on RDF-driven application development using JFace/SWT components. It provides examples of RDF-driven JFace widget, tooltip window and viewers. Each item within the scope level can be assigned a viewer, widget or container (like frames or tabs for representing a relationship of part and whole).

3.4.2 User Interface Generation

Customized UI controls are then joined together according to the provided skeleton. We did direct mapping from skeleton to graphical objects.

GUI development is an instantiation of UIO provided as skeleton. Vocabulary (with user interface properties) is instantiated as customised controls. These controls for literals (e.g. textboxes, labels, list box etc.) are arranged in group and provided to containers (e.g. Frame, widget, tab pages, web pages etc.) This structure of UI objects belonging together provides a complete GUI instance based on Domain ontology.

Taking the Garret's model in consideration, the mapping process follows the steps scope (vocabulary), structure (Domain Ontology) and skeleton (User Interface Ontology) [see Figure 3]. Skeleton provides a base design for user interfaces that can be instantiated by any graphical library for any platform.

[2] http://ibm-slrp.sourceforge.net/wiki/index.php/Com.ibm.adtech.telar.ui.swt

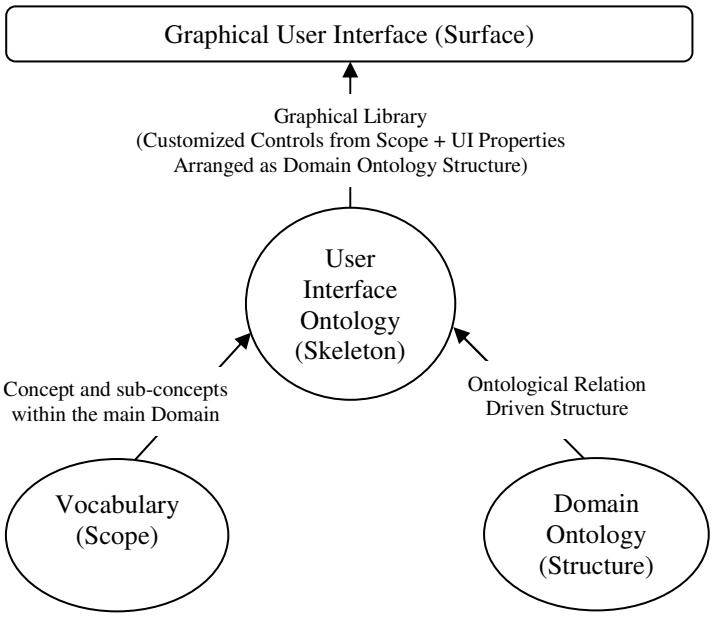

Figure 3: Ontology representation at GUI

As an example application of PIM we created a dictionary for vocabulary through reading RDF.

4 Ontology Modelling based on Context

Domain ontologies structuring and arrangement is based on relationships. In our example of vCard, properties groups and vCard groups are made on the basis of criteria provided by the whole system. vCard is based on the concept of business cards. Thus the arrangement and groups of properties will be different from a personal data ontology in social network. Individual properties (like name, address, date of birth etc.) represent the same domain ontology. These domain ontologies are arranged by different rules in an organization than in a social network.

DOLCE ontology has discussed the calculus of individuals in an organization based on their roles. The organisational structure does provide structure for grouping properties and persons.

In a network of friends, level of trust is not dependent on the official designation or department. Social networks provide different criteria to arrange and group properties and persons. All of these structures are discussed and available in relationships, axioms and calculus in current research [Bottazi 06][Gangemi 02][Clarke 81][Gruber 93]. These structures assist us to develop UI accordingly.

5 Results

This research resulted in a very simple user interface for a vCard but it was more focused on the process and methodology of UI development. There are many alternatives to represent the same concept at surface level. A group of concepts can be aggregated by a group, tab or frame. In this example we group different vCard properties by tabs as containers to show groups. The vCard can be shown in a business card view or as personal data page. Microformats also provide two views at hCard creator; the first view is a web form for data entry and the second is a preview of a vCard3. It is an example for an UI based on modes (i.e. read mode and edit mode). Microsoft Outlook also creates the same kind of UI, but with a more complex structure for vCard edit mode.

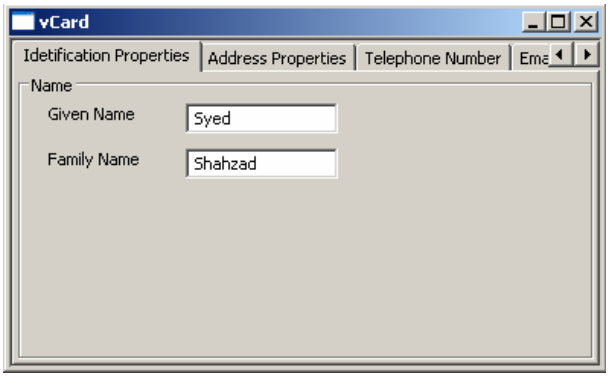

Figure 3: Example UI for a vCard

6 Conclusion

Ontological modelling and formalization used in model driven software engineering can make concept specification and representation more consistent. It helps in ensuring proper concept delivery at user interface. A proper layered approach being based on layers of user perception can make the process of GUI design and development more user-centred. User Interface Ontology is a conjunction of the domain ontology, context and look and feel. Website designs and models can also be tested at any stage through formalized concepts, as they carry the concept throughout the development process. In our approach, representation methods were specified as default for textual data. In general, knowledge representation depends on the knowledge domain specified. User interaction and response methods are based on the use and affordances of the knowledge domain. These interaction methods can be specified through ontological functions.

Future work will consider axiomatic constraints or validation functions that can be attached to a model as a part of the ontology rather than through implementation at

[3] http://microformats.org/code/hcard/creator

the GUI level. Further, user Interface properties can form a generic ontology with the UI control vocabulary and relations among them. This UIO can be fused with any domain ontology to introduce UI properties of the domain ontology.

Acknowledgements

This contribution is partly funded by the Know-Center and the Higher Education Commission of Pakistan. The Know-Center is funded within the Austrian COMET program -- Competence Centers for Excellent Technologies -- under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[Alexander, 03] Alexander K., Valeriya G.: From an ontology-oriented approach conception to user interface development. Int. J. Information Theories & Applications, Vol. 10, No.1, p. 87-94, 2003.

[Bottazi 06] Bottazzi, E., Ferrario, R.: Preliminaries to a dolce Ontology of Organizations, Int. J. Business Process Integration and Management, Vol. 1, Nos. 1/2/3, pp.64–74, 2006.

[Clarke 81] Clarke B.: A Calculus of Individuals Based on 'Connection', Notre Dame Journal of Formal Logic, Volume 22, Number 3, July 1981

[Furtado 02] Furtado E., Furtado J., Silva W.,Tavares D.,: An Ontology-Based Method for Universal Design of User Interfaces, Task Models and Diagrams For User Interface Design (TAMODIA 2002)

[Gangemi 02] Gangemi A., Guarino N., Masolo C., Oltramari A., Schneider L.: Sweetening Ontologies with DOLCE, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, p.166-181, October 01-04, 2002.

[Garrett 02] Garrett J.: The elements of user experience: user-centred design for the web, ISBN-13: 978-0735712027, October, 2002

[Gruber 93] Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing, International Workshop on Formal Ontology, 1993, Padova, Italy, March 1993

[Liu 05] Liu B., Chen H., He W.:Deriving User Interface from Ontologies: A Model-based Approach, 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), Hong Kong China

[Paulheim, 09] Paulheim H., Probst F.: A Framework for Ontologies-based User Interface Integration, The 8th International Semantic Web Conference (ISWC 2009), Northern Virginia, USA, 25-29 October 2009.

[Shahzad, 09] Shahzad, S., Granitzer, M., Tochtermann K.: Designing User Interfaces through Ontological User Model, Proceedings of the Fourth International Conference on Computer Sciences and Convergence Information Technology ICCIT 2009, Seoul, Korea, 24-26 November 2009. p 99-104.

[Smith 04] Smith, B.: Beyond concepts: ontology as reality representation. In: Proceedings of the third international conference on formal ontology in information systems (FOIS 2004). Amsterdam: IOS Press; 2004. p. 73-84.

[Uschold 08] Uschold, M.: Ontology-Driven Information Systems: Past, Present and Future, FOIS 2008, Saarbrücken, Germany, Oct 31st – Nov 2008, doi:10.3233/978-1-58603-923-3-3