Ontological Model Driven GUI Development: User Interface Ontology Approach

Syed K Shahzad
Knowledge Management Institute
Graz University of Technology
Graz, Austria
shahzad@student.tugraz.at

Michael Granitzer
Know-Center Graz
Graz University of Technology
Graz, Austria
mgranitzer@tugraz.at

Denis Helic
Knowledge Management Institute
Graz University of Technology
Graz, Austria
dhelic@tugraz.at

***Abstract*- Ontology and Semantic Framework has become pervasive in computer science. It has huge impact at database, business logic and user interface for a range of computer applications. This framework is also being introduced, presented or plugged at user interfaces for various software and websites. However, establishment of structured and standardized ontological model based user interface development environment is still a challenge. This paper talks about the necessity of such an environment based on User Interface Ontology (UIO). To explore this phenomenon, this research focuses at the User Interface entities, their semantics, uses and relationships among them. The first part focuses on the development of User Interface Ontology. In the second step, this ontology is mapped to the domain ontology to construct a User Interface Model. Finally, the resulting model is quantified and instantiated for a user interface development to support our framework. This UIO is an extendable framework that allows defining new sub-concepts with their ontological relationships and constraints.**

I. INTRODUCTION

Semantics and ontological framework defines computational model for concepts [1] [2]. This framework can provide a base for modeling concepts representation at user interface to the end users. Uschold [12], Schlungbaum [6], Liu [4], IBM[1], Shahzad [8] in their research argued for the model driven user interface development. This paper also argues for model based user interface development using semantics and ontological framework. Here, we construct ontology defining some basic user interface classes, properties and their relationships in extendable framework as User Interface Ontology (UIO). In an incremental process, UIO and targeted domain mapping presents the base user interface model. This model is quantified and instantiated to develop a Graphical User Interface (GUI).

Next section of this paper focuses at the motivation behind this research to support our arguments and our work. Third part discusses the UIO development process. Forth section provides the association of UIO with domain ontology. In final section, this paper describes one of the methods for instantiation and development of the Graphical User Interface (GUI), followed by conclusion and future work.

In this research, we use an example of vCard as a domain ontology for personal information management system, to represent at GUI. The impetus of this paper targets the area of modeling and development of a consistent graphical user interface that ensures the semantic properties and constraints defined in the domain ontology.

II. MOTIVATION

Since last decade, Semantic and ontological framework has been widely used for information retrieval and knowledge representation (e.g. Linked Open Data[2] and Web3.0[3]). Various GUI facilitate the representation and visualization of the information retrieved from semantic web at heterogeneous platforms. These GUI have been built in variety of flexible development environments, which allow interface for developers to provide visualization and user interaction mechanism that is disjoint to the semantic relationship and constraints. This phenomenon results in many inconsistent representations of the concepts at similar or different platforms.

Thus, there is a need to explore a methodology that ensures the semantic rules (relationships and constraints) at user interface level for a consistent GUI development in various environments. There is also a need to use these rules for defining data validators, information visualization and user interaction techniques at user interfaces level.

III. ONTOLOGY ENGINEERING FOR USER INTERFACE ONTOLOGY

Here we discuss in detail about ontology engineering for UIO. In our previous work [8] [9], we introduced the term of UIO as user interface aspects of computer applications and their mapping to domain ontology.

In this work, we build UIO as UI specification to maintain the qualities of UI specification and to ensure proper concepts of user interfaces and targeted domain in GUI development process.

A. Modeling User Interface Aspects

Silva [10] defines model-based user interface development technology to provide an environment where developers can design and implement user interfaces (UIs) in a systematic

[1] http://www.ibm.com/developerworks/library/w-berry/

[2] http://linkeddata.org/

[3] http://www.w3.org/2001/sw/

Authorized licensed use limited to: Vrije Universiteit Amsterdam. Downloaded on June 21,2024 at 15:42:17 UTC from IEEE Xplore. Restrictions apply.

way. He also stated three qualities of UI specifications, such as:

1. To model user interfaces using different levels of abstraction;
2. To incrementally refine the models;
3. To re-use UI Specifications

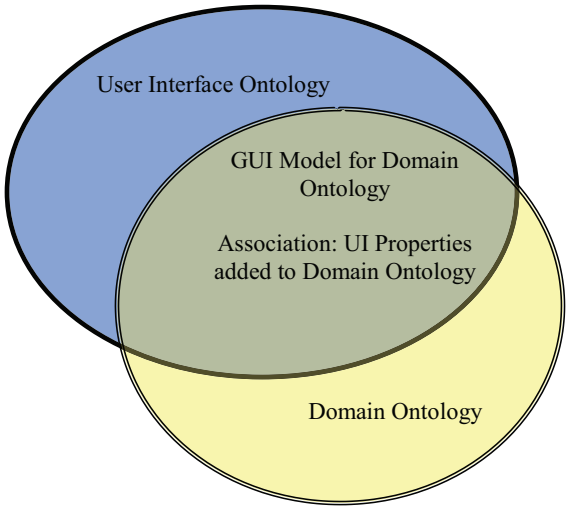

Fig1. Association: Relating Domain ontology with User Interface Ontology

In UIO, User Interface will be the domain concept to be presented through ontology. All of the user interfaces are composed of same GUI controls. End User Interfaces are much more dependent on the targeted domain ontology of the software.

B. Ontology Engineering for User Interfaces

In this stage, we have defined UIO at different level of abstractions to deal with the first property defined by Silva [10]. The Xerox Palo Alto Research Center [16] and IBM's[4] "*The iceberg analogy of usability*" discuss three levels of abstraction of user interfaces. Moreover, nearly similar but detailed levels of abstraction for user interfaces have been defined in Garrett's [15] "*The elements of user experience*". Garrett's research has been used as level of abstraction for user interface in our previous research [9]. In our approach, we have defined nearly same levels of abstraction for user interface modeling. We have added context dependent properties which can be quantified at the GUI development process. These levels are:

1. Data Modeling
   a. Data Formats
   b. Data Structures (Information Architecture)
2. User Interaction Properties
3. Graphical Properties
4. Context aware properties

Data Modeling: Data modeling classes are defined as purely abstract to the implementation details. It specifies only the data domain and its hierarchy. For the research purpose we deliver a paradigm based on ontological classes for data formats from the set controls and user events applied in java (SWT[5]). It provides a base data model which has to be represented at user interface with specific user interaction methods and graphics detail.

Classes and sub classes made for data format and data structures criterion:

1. Visualization Classes:
   a. Group
      i. Widget
      ii. Frames
      iii. Tabs/Pages
   b. Textual
   c. Temporal
   d. Multimedia
      i. Picture
      ii. Audio
      iii. Video
2. Data Structure based Classification:
   a. Single Entity
   b. List
   c. Tuple
   d. Table-Grid

Visualization classes characterize the representation methodology of a concept, depending of its semantic class and in our example RDF/XML data type.

Data Structure based Classification specify the information architecture at user interfaces. Mereo-toplogy also helps user interface designing by providing the relations to give a consistent representation of concept in their semantic groups.

User Interaction Properties: At lower level of abstraction to data modeling classes, we have defined user interaction properties. These properties are dependent to the data modeling classes which specify the user interaction methodology for a specific data model in general or in specific architecture.

Graphical Properties: At next level of abstraction to user interaction properties, we have defined graphical properties. These properties are also dependent at data modeling classes.

Context aware Properties: At the lowest level of abstraction, we defined context aware properties. These properties are also dependent at data modeling properties and user environment specification. This specification consists of technological details and user information.

This ontology was made in OWL-DL and RDF as meta-data for the GUI. There were also some rules defined as basic user interface rules that specify relationships. For instance, a Group is a container that is a set of other classes like textual-data, lists and/or other groups. UIO also provides constraints that ensure consistency at user interface, for instance e.g. duration property (property of audio/video format) cannot be associated to textual data. This ontology is extendable to add more features and aspects related to UI.

4 http://www.ibm.com/developerworks/library/w-berry/

5 http://www.eclipse.org/swt/docs.php

Authorized licensed use limited to: Vrije Universiteit Amsterdam. Downloaded on June 21,2024 at 15:42:17 UTC from IEEE Xplore. Restrictions apply.

IV. MAPPING DOMAIN ONTOLOGY WITH UIO

UIO mapping to domain ontology is performed in two major steps. Firstly, data modeling classes from UIO are associated with the domain ontology properties to provide a structure for information visualization and architecture. Secondly, user interaction and graphical properties from UIO are added to the mapping. Any user interface concept specified in UIO is linked to the domain ontology properties in three ways:

1. Link to all properties having same range
2. Link to all properties having same domain
3. Link to the specific relation (Reification)

This mapping proceeds through the UIO levels of abstraction from top to bottom.

A. Mapping Visualization Classes

At the first step of mapping process, representation based classes are mapped to the properties of target domain ontology to specify the visualization method. It is mostly dependent on the range class of the entity that specifies the data format, but the procedure also allows mapping based on domain or to the specific relation of the domain and range.

All next level properties are mapped according to the representation specified in visualization class mapping to any domain ontology property.

B. Mapping User Interface Properties

User interface properties are mapped to domain ontology properties depending on their visualization classes, e.g. Editable is a user Interface property for Textual data, while multi-selection can be a property of list or a Table-Grid. UIO provided rules ensure semantic association of user interface properties to every data type. UIO also ensure data consistency at quantifying these properties e.g. if a container is disabled for user interaction and performing any operation all entities within the container will be disabled as well. Rather than depending at graphical libraries, semantic rules are defined in UIO and carried to GUI for consistent representation.

C. Mapping Graphical Properties

Graphical properties are mapped to domain ontology for providing a structure of interface entities. It can also map properties which can be quantified at the time of GUI development. These properties has to provide relative scaling of user interface entities in mapping process e.g. font sizes for title, headings and body text. These relative scaling collectively provide a theme or representation styles.

D. Mapping Context aware properties

These properties are mapped to the domain ontology properties but can be quantified only at GUI development process.

E. User InterfaceModel (UIM)

This mapping of DO and UIO results in a User Interface Model (UIM). This Model is base for the user interface generation. Similarly, the main target of this research is also to create such a technology and context independent model for any given domain concept representation.

Semantic knowledge representation model contains a combination of semantic rules (relationship and constrains) from UIO and domain ontology that ensure:

1. Proper concept representation at GUI
2. Consistent concept representation
   a. By GUI developed with different technologies
   b. At heterogeneous platforms

F. vCard Ontology for Personal Information Management

In this stage, we continued working on the same Personal Information Management case study/example [9]. vCard is a standard and widely adapted (Apple's Address book, MS Outlook, TUGraz Staff and Student[6], Mobile Phones) ontology. It is defined by Internet Mail Consortium (IMC)[7] and Personal Data Interchange as vCard (RFC2426). This paper focuses on factors and issues in mapping of vCard to GUI. vCard namespace[8] provides a vocabulary and schema of a vCard. This schema can be associated with UIO according to the RDF/XML data type entities

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

Table 1: Structure of vCard properties [9]

Apparently, the presented model for semantic description of associations looks complex, however, the ontology engineering software will not allow making any inconsistency in ontology. Introducing new relations and adding new relations can only be done by HCI expert and Ontology engineer. We have done manual association in this study, such as associating First name with a textual class. It is like making a relations or RDF statement for each relations. We have added UI aspects as objects like

[http://www.w3.org/2006/vcard/ns#Name, Displayby, "UIO/ns#Textual"]

[6] https://online.tugraz.at/tug_online/visitenkarte.show_vcard?pPersonenId=73A4CED7D8FE1923&pPersonenGruppe=3

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

[8] http://www.w3.org/2006/vcard/ns#

Authorized licensed use limited to: Vrije Universiteit Amsterdam. Downloaded on June 21,2024 at 15:42:17 UTC from IEEE Xplore. Restrictions apply.

Futhermore, some UI aspects were added as to be quantified at later stage such as:

[http://www.w3.org/2006/vcard/ns#Name, Font, "0"]

V. USER INTERFACE GENERATION

A. *Quantifying Context aware properties*

Before proceeding to the instantiation we need to quantify the UI aspects which are dependent on the context or technology. Introduction of context also make the user interface sketch more detailed and concrete.

These context variables depend at the device for which GUI is going to be made e.g. iPhone, desktop or web application. Information architecture, and navigation remains same for this device but size and wrapping can be different depending on the context.

B. *Instantiating UIM*

All Domain ontology associated with UIO that come up with a complete sketch of user interfaces in RDF/XML format (UIM). Any graphical library can be used to instantiate the UIM. We have used SWT (Java) for UIM instantiation of vCard.

By using RDF/XML, UIM facilitates to represent the data in a simple text format without making any GUI.

VI. UIO IN SOFTWARE DEVELOPMENT ENVIRONMENT

This model also separates the individual and collaborative work knowledge experts. It is also an economical solution for software development firms. However, every firm cannot bear the cost of HCI consultants for each of the software development. There are mostly reusable libraries are made for some specific domain. That does not fit or get associated with every domain. Secondly, all HCI concerns are done only for customized or specific software.

1. In Fig2 HCI expert build at B. UIO experts and

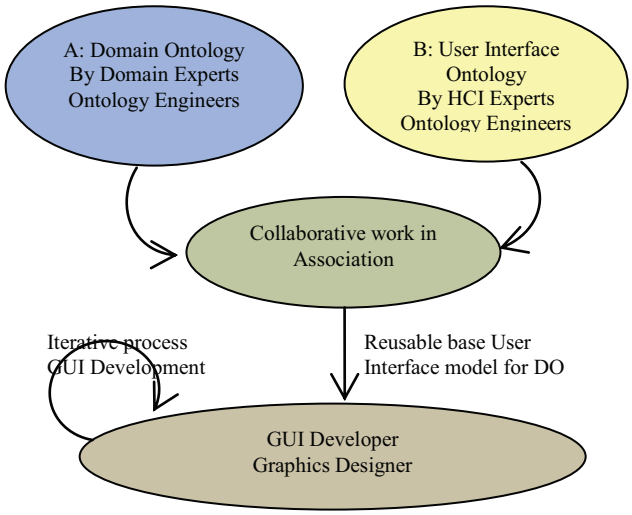

Fig2: Knowledge Experts contribution breakdown and collaboration stages

Domain expert can work at A. These ontology are

   a. Independent of context details
   b. Independent to any technical and implementation details
   c. Reusable for association with any domain
   d. Extendable with new vocabulary
   e. Shared, Standardized also extendable by other skilled resources
2. At association level a sketch can be made for a user interface
   a. Independent to the technology details
   b. Context can be introduced
   c. Association is done at this layer considering basic requirement of each knowledge domain
   d. Provide reusable User Interface Model (UIM) as a base for GUI development for the associated domain ontology
3. At GUI generation level GUI developer and graphics complete sketch of the picture which they can colour with any graphical library, themes style-sheets etc. At the end a GUI is instantiated from this level which
   a. All Technical aspects are considered rather than design aspects
   b. Multiple GUI can be created with the same sketch for different devices which keeps a consistency.

VII. RESULTS

This research focused at process of user GUI and consistent concept delivery to the end user through GUI. The end result of the process is just a normal GUI for a vCard, but main targeted result is UIM - stated as RDF/XML files containing the tags of domain ontology and UIO. It is like a complete ready recipe just to put in machine to start working. Further results show that UIO mapping process read vCard schema from RDF file. The intelligence of this research was UIO engineering and mapping of UIO that can be applied to any domain ontology.

VIII. CONCLUSION

Rather than using raw user interface development guideline, it was evident from our work that user interfaces properties and their relationships can be defined through semantic and ontological framework as UIO. We also illustrated that UIO mapping with any domain ontology gives an abstract base model for user interfaces for the domain ontology. The long term goal of this paper is to demonstrate a model guide and to restrict user Interface developer to develop any semantically consistent GUI for heterogeneous development and execution environment.

IX. FUTURE WORK

It is increasingly observed that research results usually come up with new questions. Similarly, this work attempted to recommend some approaches and model to the development of ontological model based user interface, however, many questions still have to be explore in this domain. We are still working on to use UIO for different standard domain ontology which is extending UIO for many new formats of data and

Authorized licensed use limited to: Vrije Universiteit Amsterdam. Downloaded on June 21,2024 at 15:42:17 UTC from IEEE Xplore. Restrictions apply.

properties and their relationships e.g. multimedia properties, vector data. There are more areas to explore in this regard such as:

- There is a big question of usability for resulting GUI which will explored in future research.
- There is also need to discuss functional ontology for domain ontology and UIO (user actions and events).
- There is also a need to make some base UIO and browser which can read and display UIM.
- AI and Interactive learning from end users can make a mechanism for auto extend and align UIO.

X. ACKNOWLEDGMENT

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. Invader

REFERENCES

[1] N. Guarino, "Formal ontology, conceptual analysis and knowledge representation", *International Journal of Human and Computer Studies*, vol43(5/6): pp 625-640.1995

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

[3] Horridge M., Jupp S. "A Practical Guide to Building OWL Ontologies Using protégé and Co-ODE tools" *Book Edition 1.1*, The University of Manchester, October 2007.

[4] 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

[5] Myers B. "User-Interface Tools: Introduction and survey User-Interface Tools: Introduction and Survey" *IEEE Software*, v.6 n.1, p.15-23, January 1989, doi:10.1109/52.16898.

[6] Schlungbaum E. "Model-Based User Interface Software Tools – Current State of Declarative Models" *Technical Report* 96-30, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 1996.

[7] 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.

[8] 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.

[9] Shahzad, S. "Ontology-based User Interface Development: User Experience Elements Pattern" *Journal of Universal Computer Science,* vol. 17, no. 7 (2011), 1078-1088

[10] Silva P.: User Interface Declarative Models and Development Environments: A Survey, Proceedings of the 7th international conference on Design, specification, and verification of interactive systems, p 207-226.

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

[12] Uschold, M. "Ontology-Driven Information Systems: Past, Present and Future" *Proceedings of the seventh international conference on formal ontology in information systems (FOIS 2008)*, Saarbrücken, Germany, Oct 31st – Nov 2008, doi:10.3233/978-1-58603-923-3-3

[13] Paula M., Barbosa S., Lucena C. "Conveying Human-Computer Interaction Concerns to Software Engineers Through an Interaction Model", *Proceedings of The second Latin American Conference on Human-Computer Interaction, CLIHC'05*, October 23-26, 2005, Cuernavaca, México. ACM 1-59593-224-0, P 109-119.

[14] Almeida M., Souza R., Fonseca F. "Semantics in the Semantic Web: A Critical Evaluation", *Knowledge Organization Journal*. Vol. 38, n.3 p. 187-203. 2011.

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

[16] K. Alexander, G. Valeriya. "From an ontology-oriented approach conception to user interface development". *International Journal Information Theories & Applications*. Vol. 10, num.1, p. 87-94, 2003.

Authorized licensed use limited to: Vrije Universiteit Amsterdam. Downloaded on June 21,2024 at 15:42:17 UTC from IEEE Xplore. Restrictions apply.