Capturing Semantic Meaning on User Interface Presence By Creating Its Ontology

Elviawaty M. Zamzami[1,2] and Eko K. Budiardjo[2]

[1] Department of Computer Science, University of Sumatera Utara
Medan, 20155, Indonesia

[2] Faculty of Computer Science, University of Indonesia
Depok, 16424, Indonesia

Abstract

Screenshots, known for years as capturing Graphical User Interface by means of print screen button or capturing software, is not only beneficial for creating user guide or user manual document, but also for reverse engineering process. This paper presents a new way of capturing the data appear on a screen by creating the ontology of its interface. Data capturing based on the interaction style of interface or windowing system, known as WIMP (Window Icon Menu Pointer). The Ontology model used as template, called as WIMP-UI. OWL is used as ontology language, Portégé as editor tools, and Pellet reasoner for reasoning.

Keywords: *User Interface, Interaction Style, WIMP Interface, Ontology, WIMP-UI Ontology.*

1. Introduction

Creating user manual, on screen user interfaces are heavily captured during this activity, resulting comprehensive manual. As a guidance, the manual contains information in which beneficial for the user to understand its detailed interaction between user and computer. By understanding this information, appropriate and ease of use will increase the software usage comfortability.

Generally, software engineer captures screen interface using software service, such as *print screen*, *screenshots*, *FastStone Capture*, and the others. This paper conveys about an alternative way to capture user interface screen. This alternative way is creating an ontology. Using ontology, screen interface is not represented as picture but as formal representation. This ontology represents data on user interface presence. Creating ontology in this paper using OWL as ontology language, Portégé as editor tools, and Pellet reasoner for reasoning. Portégé can also help software designer (programmer) by means provide some services to generate code, such as schema code, or Portégé and OWL code based on ontology that created. This will not be obtained if the screens are captured as images.

2. User Interface

The user interface is the part of the software that used by user to see, hear, talk, and touch the software [6]. The user interface includes the idea of the designer about the user, the user needs and expectations, and find the requirements through artefact interactive [9].

As part of software, user interface takes a relative large portion of the overall code. With regard to the user interface was informed that [10]:

a. In 1992, 50% of application code estimated to be user interface code.
b. In 2006, 88% of GIMP's 2,000 files have dependencies on the UI toolkit.
c. 33% of Java 1.5 ".java" source is in a user interface package.

Creating user interface consume the most expensive part of software system development. User interface range 50% to 70% of software development effort total [8]. The information that contain about user interface and its elements can ease the software engineer tasks for further developments of the software.

The user interface is needed by the user and the software to interact, in which users take actions and the software takes responses that caused by the action. The interactions are done to be achieved the software goal.

3. Interaction Style

The way of user communicate to software is known as interaction style [2]. Interaction style refers to all the ways

that can be done to interact between user and software. There are several interaction style, namely:

a. Command line, the style that expresses user instruction directly using function keys, single character, abbreviation, whole words, or a combination.
b. Form-fills and speadsheets, the interaction style provides form or spreadsheets where the user can enter and alter data.
c. Menus, the interaction style that using set of options displayed on the screen using mouse, numeric, or alphabetic keys.
d. Natural language, the style that enable the user interact using natural language.
e. Point and click, the interaction style that points anything on screen and user can click its.
f. Question-answer and query dialog, the style that provide interaction mechanism where user accepts set of questions and take responses as answers in certain domain.
g. Three-dimensional, the interaction style that provide anything in 3D perception. The users use 3D input to interact [1].
h. WIMP, the interaction style that involve Window, Icon, Menu, and Pointer (WIMP) to interact between user and software.

Each of interaction style is supported by the appropriate interface. Thus, for the above interaction style, there are several types of interfaces. There are command line interface, form-fills and spreadsheets interface, menus interface, natural language interface, point and click interface, question answer and query dialog interface, three dimensional interface, and WIMP interface.

In this paper, users interact to a software using the WIMP interaction style. In such a way so that the interaction is done throughout the WIMP interface.

4. WIMP Interface

In the modern Human and Computer Interaction (HCI), interactions are dominated using Graphical User Interfaces (GUIs), also known as WIMP interface [12]. WIMP interface provides convenience to the users to learn and use it to interact with software [7], e.g. symbol ✂ as "cut" symbol. This is the reasons for choosing WIMP interface as object that researched.

The WIMP interface consists of elements like in Table 1, there are:

a. Window, a screen area that run its program independently, isolated from the other windows. Some systems allow cascading windows. Window may contain titlebar, scrollbar, toolbar, push button, radio button, check box, spin box, list box, combo box, dialog box, label box, text field, text area, table, figure, hyperlink, and palette.
b. Icon, picture with tiny size that used to represent symbolically of object and process on the screen.
c. Menu, the choice of operations or services, based on text or icon to be executed by the system. Menu may contain drop-down menu, pop-up menu, and cascading menu.
d. Pointer, symbol that represent motion of input devices to point and choose element on screen.

Table 1: Elements of WIMP user interface.

Element		Description
Window		The screen area that run its program independently, isolated from the other windows. Some systems allow cascading windows.
Window	*Title bar*	The interface component that contains of window name, or application name that currently in use.
	Scrollbar	The interface component that is used to shift screen vertically or horizontally.
	Toolbar	The component of interface where to put button, menu, and icon.
	Push Button	The interface component that present set of choice by push or click the button.
	Radio button	The interface component that allow the user choose a choice, the choice have a value 'on' or 'off'. Radio button can be grouped, but only one choice can be choosed.
	Check box	The interface component that provide some choices. User can choose more than one choice or even all of choices. Check box can be grouped.
	Spin box	The component of interface to arrange value of variable, usually is numerical type, added or reduced by the certain value. In the spin box, maximum and minimum values should clearly, so not spinning continously.
	List box	The interface component that display a number of choice and can be seen directly by software user. After user chooses one of choices, then the choice will be saved as a value of a variable.
	Combo box	The component like as list box, but have differences in appearance. In the combo box, the choices are not visible except by pressing the arrow attached to the combo box.
	Dialog box	The component that use by system to present the important information, for examples, error messages and warning messages.
	Label box	The component of interface to present comments or name of interface components.
	Text field	The interface component as a row of data field (a single line area), entered using keyboard.

Table 1: Elements of WIMP user interface (continued).

Element		Description
Window	*Text area*	The area as interface component that can be loaded by some text lines.
	Table	The interface component to load tables.
	Figure	The interface component to load figures.
	Hyperlink	The interface component that shows the object on screen that has link to the other object.
	Palette	The interface component as place that load the icons where represents the possible modes and visible by user.
Icon		The picture with tiny size that used to represent symbolically of object and process (e.g. command, file, folder, or device) on the screen.
Menu		Menus provide choice of operations or services, based on text or icon to be executed by the system.
Menu	*Pulldown/ Drop-down Menu*	Menu that drops down when user click menu title on menu bar.
	Pop-up/ Context/ Shortcut Menu	Menu that related with the object that chosen currently. Usually, with the right click of mouse on the active window.
	Cascading Menu	Menu with hierarchy structure where if chosen on the menu will open the new menu.
Pointer		The interface component as certain symbol that represent motion of input devices (e.g. mouse, joystick, trackball, keyboard) to point and choose element on screen.

5. Ontology

The user interface is usually captured in visual representation. This representation provides ease of understanding to the user interface that provided by software, but, just easily read by humans rather than by machines (computers). Creating ontology of those interface result a representation that can be read by humans and machines.

The ontology is a formal specification of a shared conceptualization of domain that attracts attention. Formal refers to the fact that the ontology should be read by the machine [4]. The shared conceptualization means that ontology captures consensual knowledge in a group [3]. Also, the ontology provides vocabulary that relate to software application domain [5].
The ontology describes basic concept in a domain and defines relation among them. An ontology contains:

a. *Class* or concept.
b. Properties of concept that describe features and attributes of concept (*slots*, also known as *roles* or *properties*)
c. Restriction of *slots* (*facets*, also known as *role restrictions*).

To obtain the detail ontology, accurate, consistent, and sound, the ontology can be expressed using a logic-based language. [11]. This paper uses OWL-DL as the ontology language.

6. WIMP-UI Ontology

Ontology in this paper will represent the user interface, referring to interaction style. Ontology is developed using ontology model as template, called as WIMP-UI (Window Icon Menu Pointer – User Interface). Data on user interface presence are captured and transformed as elements of the WIMP-UI ontology.

The WIMP-UI ontology is reviewed from the depth of ontology model, has four (4) levels. The first level, represents the user interaction that requires the interaction style to involve the users interact with the software using the interface. The first level described in the following Figure 1:

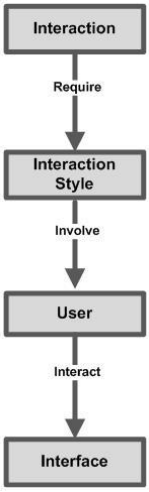

Fig. 1: The 1st level of WIMP-UI ontology.

Further, in the second level represents type of interface styles based on the interaction style. The second level of WIMP-UI ontology is presented in Figure 2.

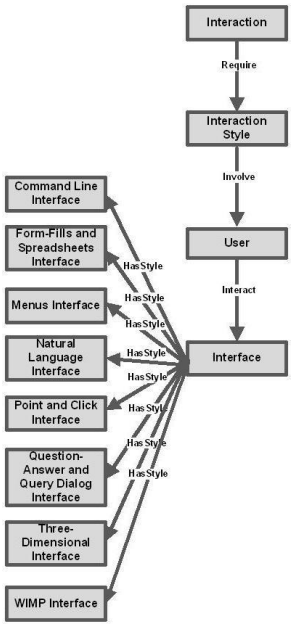

Fig. 2: The 2nd level of WIMP-UI ontology.

Then, in the third level is represented elements of the WIMP interface. The 3rd WIMP-UI ontology is represented in Figure 3.

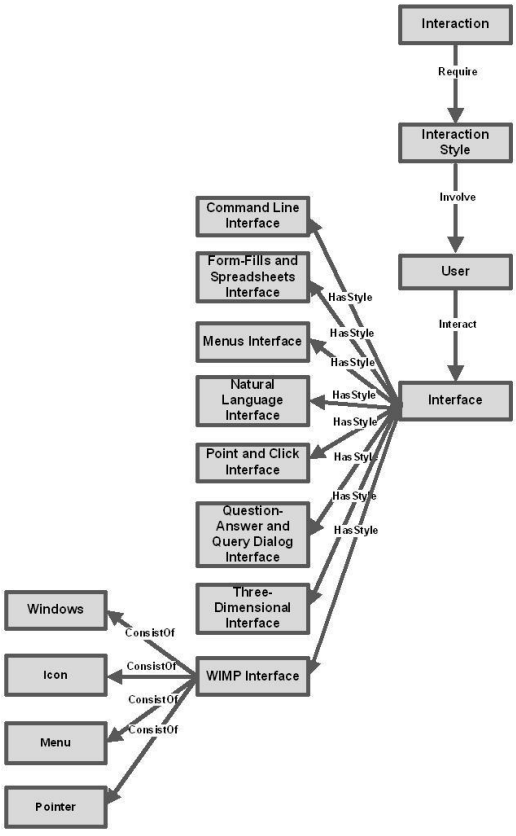

Fig. 3: The 3rd level of WIMP-UI ontology.

The fourth level represents the WIMP interface in detail. The 4th level is the last level, represents the elements contained in the WIMP interface elements, described in Figure 4 below:

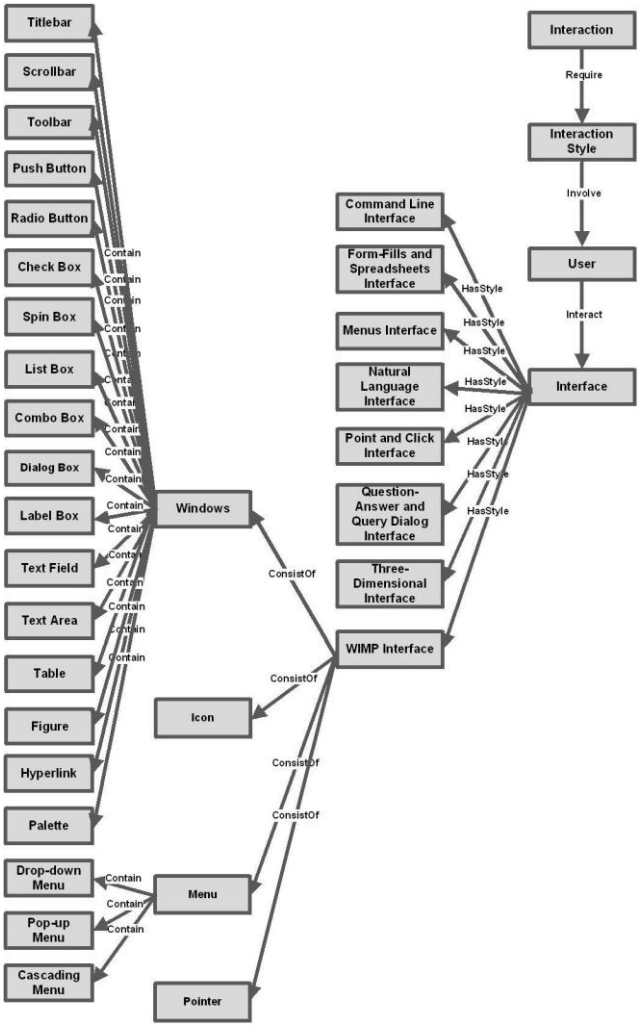

Fig. 4: The 4th level of WIMP-UI ontology.

Description of the elements of the WIMP-UI ontology can be seen in Table 2.

Table 2: Elements of WIMP-UI ontology.

Element Type	Element of WIMP-UI Ontology	Description
Class	*Interaction*	Communication that occur between user and software.
	Interaction Style	The nature of interaction.
	User	A person (a group of person) that interacts with software to achieve user goal.
	Interface	Part of interactive computer system that sends messages to the user and receiving instructions from the user.

Table 2: Elements of WIMP-UI ontology (continued).

Element Type	Element of WIMP-UI Ontology		Description
Class	*Command Line Interface*		The interface that expresses user instruction directly using function keys, single character, abbreviation, whole words, or a combination.
	Form-Fills and Spreadsheets Interface		The interface that provides form or spreadsheets where the user can enter and alter data.
	Menus Interface		The interface that provides set of options displayed on the screen using mouse, numeric, or alphabetic keys.
	Natural Language Interface		The interface that enable the users interact using natural language.
	Point and Click Interface		The interface that points anything on screen and user can click its.
	Question-Answer and Query Dialog Interface		The interface that provides interaction mechanism where user accepts set of questions and take responses as answers in certain domain.
	Three-Dimensional Interface		The interface that provides anything in 3D perception.
	WIMP Interface		The interface that consist of Window, Icon, Menu, and Pointer as its components to interact between user and software.
	Windows		contained in Table 1
	Windows	*Title bar*	
		Scrollbar	
		Toolbar	
		Push Button	
		Radio button	
		Check box	
		Spin box	
		List box	
		Combo box	
		Dialog box	
		Label box	
		Text field	
		Text area	
		Table	
		Figure	
		Hyperlink	
		Palette	
	Icons		
	Menus		
	Menus	*Pulldown/ Drop-down Menu*	
		Pop-up/ Context/ Shortcut Menu	
		Cascading Menu	
	Pointers		
Property	*Require*		*Domain* requires *range*.
	HasStyle		*Domain* has style, a *range*.
	UsedBy		*Domain* is used by *range*.
	Involve		*Domain* involves *range*.
	Interact		*Domain* interact to *range*.
	ConsistOf		*Domain* consist of *range*.
	Contain		*Domain* contain *range*.

7. Procedures of Capture and Transform

Capturing data on user interface presence by creating the WIMP-UI ontology should be done conform to procedure of data capture and procedure of data transform to be WIMP-UI ontology. The procedures of data capture are:

a. Observe all of data on user interface presence to determine elements that provided by software.
b. Observations are done for each windowing that appear on the user interface screen.

Each of elements that found as observation results are transformed using template of the WIMP-UI ontology. Procedures of data transform are:

a. Create *sub class* of `WIMP Interface` *class*, be named as `WIMP Interface-1..n`. The *sub class* contains all of *windowing* that exist in the software, from *window-1* until *window-n*. The procedure is done as first step of interface transformation and only once.
b. Create *sub class* of `WIMP Interface-1..n`, be named refer to the format `WIMP Interface-n`, where `n`=1..n. Example, if be named as `WIMP Interface-3`, it represents all of the elements that exist in the third *windowing*. After this, procedure c till to e are done to transform each element that observated exist in *windowing-3*.
c. Check the existence of interface elements that observed as *sub class* of *class* of the `Window`, `Icon`, `Menu`, or `Pointer`.
d. If not found then create *sub class* of *class* of the appropriate element. Name of *sub class* refer to `<Window|Icon|Menu|Pointer>-<text|label|that appear in the element on screen>`. For example, if on the screen interface appears the element interface which be included as *icon* and has label *cut*; then *sub class* be named as "`Icon-Cut`".

e. For all of the *sub class* that added, should be given a brief description in "`rdfs:comment`".
f. Create restriction of `ConsistOf` property and the inverse property (`PartOf`) for `WIMP Interface-n` and all of classes that represent the elements of `WIMP Interface-n`.
g. Create restriction of `Interact` property and the inverse property (`InteractedBy`) that relates sub class of `user` and sub class of `WIMP Interface-n` that used by user.

For example, there is screen of software interface like as Figure 5. Observation is done against the elements of interface, the results obtained:

a. *Text area* that contains information about authentication errors.
b. *Text area* that contains information about the active period of user account.
c. *Text area* that contains information about faculties telephone number in University of Indonesia.
d. *Text area* that contains information about directorate of finance in University of Indonesia.
e. *Hyperlink* to https://academic.ui.ac.id.
f. *Hyperlink* to https://profile.ui.ac.id.
g. *Text field* to enter *username*.
h. *Text field* to enter *password*.
i. *Push button* to enter *Login* command.

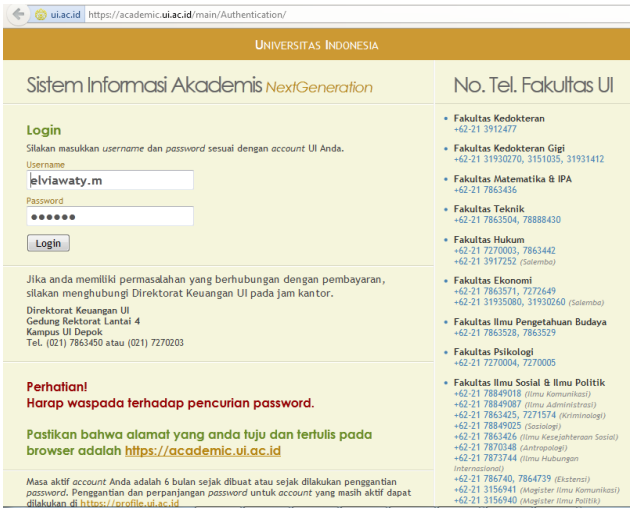

Fig. 5: The example of interface screen that observated.

Then, the data captured are transformed into ontology template (WIMP-UI) refer to the procedure of data transformation. Using Portégé can be visualized as in Figure 6.

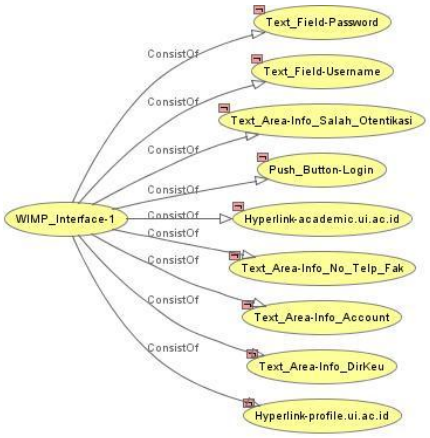

Fig. 6: The example of interface screen that observated.

8. Conclusions

From the above discussion and research conducted by the author, it is conclude as follows:

a. User interface can be captured using an alternative way, WIMP-UI ontology, ontology that represents elements of user interface
b. Using ontology, data on user interface that captured can be read by human and computer, also expected can support software development toward the existing user interface.
c. Completeness of the WIMP-UI ontology highly depend on the results obtained by the observer to represent the actual user interface.

References

[1] Bowman,Doug A., et al, "3D User Interface: New Direction and New Perspectives", Computer Graphics and Applications, IEEE, Vol.28, Issue 6, 2008, pp.20-36.
[2] Dix,Alan, Finlay,Janet, Abowd,Gregory D., Beale,Russell, "Human-Computer Interaction", 3rd Edition, Pearson Education Limited, Prentice Hall, 2004.
[3] Ehrig,Marc, and Sure,York, "Ontology Mapping – An Integrated Approach", The Semantic web: Research and Applications, Springer, 2004.
[4] Fensel,Dieter, "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce", Springer, 2001.
[5] Havlice,Z., et al., "Knowledge in Software Life Cycle", IEEE, 2009.
[6] Lauesen,Soren, "User Interface Design: A Software Engineering Perspective", Pearson Education Limited, Addison Wesley, 2005.
[7] Molina,José Pascual, et al., "Bridging the gap: developing 2D and 3D user interfaces with the IDEAS methodology", available at http://virtual.inesc.pt/dsvis03/papers/08.pdf , Springer, 2003.

[8] Paulheim,Heiko, "Ontology-based Application Integration", Springer, 2011.
[9] Prates,R., De Souza,C., and Barbosa S., "A Method for Evaluating the Communicability of User Interfaces", Magazine interactions, ACM, Vol.7, Issue 1, 2000.
[10] Terry,Michael, "CS 349/SE382 User Interface", January 5, 2009, available at http://hci.uwaterloo.ca/courses/ CS349/ w09/lectures/lecture_1_intro.pdf , 2009.
[11] W3C, "OWL Web Ontology Language Use Cases and Requirements", available at http://www.w3.org/ TR/webont-req/, 2003.
[12] Wu,Yongmei, "Design Patterns and Frameworks for Developing WIMP+ User Interfaces, Department of computer science, Technical University of Darmstadt, Germany, 2001.

Elviawaty M. Zamzami graduated from Bandung Institute of Technology (Indonesia), magister of informatics, 2000. Student of doctoral of computer science at University of Indonesia, 2009 - now. She is a lecturer at department of computer science, University of Sumatera Utara, Indonesia. Current research interests in reverse engineering, requirements recovery and ontology.

Dr. Eko K. Budiardjo has been the faculty member of the Faculty of Computer Science - University of Indonesia since 1985. Teaching, research, and practical services are aligned, resulting a full spectrum of academic achievement. Majoring in Software Engineering as professional track record, he has made some scientific contribution such as Software Requirement Specification (SRS) patterns representation method and Front CRM Framework. Graduated from Bandung Institute of Technology (ITB) in 1985, holds Master of Science in Computer Science from the University of New Brunswick – Canada in 1991, and awarded Philosophical Doctor in Computer Science from the University of Indonesia in 2007. Currently he is one of The National Research Council (DRN) member on ICT Commission, member of ICT Evaluation Working Group of The National ICT Council (DetikNAS), and Chairman of IPKIN.