Proceedings of the 4th International Conference on Information Technology and Multimedia at UNITEN (ICIMU' 2008), Malaysia
17th – 19th November 2008

Conceptual Modeling in Web-based Hypermedia Engineering with Com+HDM

Azrul Hazri Jantan
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor.
azrul@fsktm.upm.edu.my

Putra Sumari, Shahida Sulaiman
School of Computer Sciences, Universiti Sains Malaysia, Penang.
{putras, shahida}@cs.usm.my

Abstract

The coexistence of different complex features in web hypermedia applications such as information structures, navigation access mechanisms, and user interface interaction, raises new issues on the user-centered modeling approach paradigm. Thus, to provide a clear structured view of application domain, web designers must identify and clearly define complex processes existed in such applications. In this paper, we focus on conceptual design stage for complex processes in web hypermedia applications. We introduce a hypermedia design method called Com+HDM which uses UML profiles and its notation as the main graphical notation in the modeling stage. Com+HDM deals conceptual design through its three sub-design stages, called (a) conceptual-class design, (b) complex-process flow design, and (c) conceptual-process design. We present new UML specific modeling elements (stereotypes), which are defined according to the general extension mechanism in UML profile to support complex processes modeling. Each stereotype reviewed will be elaborated clearly on its definition and functionality. To illustrate our modeling example, a case study of SLEX-Web is provided. The core modeling concepts are discussed and summarized at the end of the paper.

1. Introduction

The processes of designing and developing web hypermedia applications have raise a number of design issues and increase the challenges on hypermedia modeling approaches [5]. Unfortunately, the complexity of these applications is often deceptive [14], making the application fails in their functionalities, lack of qualities, and struggling to meet user requirements [1,15]. According to [4], web hypermedia applications should support variety form of information structures, high quality of user interface, complex application processes, and navigation access mechanisms. Thus, a number of modeling concepts have been proposed in current hypermedia design methods, such as Relationship Management Methodology [6], Object-oriented Hypermedia Design Method [7], Website Design Method [8,9], UML-based Web Engineering [10,11], and Object-oriented Hypermedia [12]. However, a comparative study has been taken in [21] that show more design efforts are required to take place for supporting more complex structures of web hypermedia applications. We define *complex structures* as the following terms –applications that comprise of nested processes (which contains set of sub-processes / subtasks); applications with processes that can be suspended or resumed again (depends on user demand); and applications that enforce processes to be synchronized within their predefined activity sequence.

One of our main objectives in this paper is to present Com+HDM – *Comprehensive Hypermedia Design Method for Complex Processes Modeling*. It is a UML-based and uses UML profiles for modeling functional requirements and three major design processes, namely *conceptual design*, *navigational design*, and *user interface design* [1]. Due to the limitation space in this paper, we only focus our discussion on the *conceptual design* stage. Our main concern is three folds, (a) to describe the application domain through class diagram, (b) to capture flow of complex processes through activity diagram, and (c) to present complex processes through the definition of extended UML profiles (stereotypes). We provide a case study of SLEX-Web – *Self-paced Learning with Exercise on the Web* as our running example to show how our modeling concepts suits into the complex modeling paradigm.

This paper is organized as follows. Section 2 introduces a brief description of Com+HDM and its methodological approach. In Section 3, we briefly describe *requirement analysis* to capture functional requirements of web users and present the result in use case models. Then we describe in details two design activities of conceptual design, called *conceptual-class design* and *complex-process flow design*. Our main contributions of this paper are described in Section 4. We propose and define a number of UML specific modeling elements, called *stereotypes* to capture and model flow of complex processes in hypermedia applications. A new model, '*conceptual-process model*' (CPM) is proposed and constructed to present the final result of complex processes modeling. Finally, we conclude and summarize our work at the end of the paper.

2. Com$^+$HDM: A Modeling Approach

This paper concentrate on a hypermedia design method called Com$^+$HDM – *Comprehensive Hypermedia Design Method for Complex Processes Modeling* to support complex web hypermedia application design and modeling. We put our special focus on the systematic conceptual design, navigation modeling paradigm, and user interface interaction. In general, Com$^+$HDM is an object-oriented approach based on Unified Modeling Language (UML) profiles [13] and has similarity to existing UML-based Web Engineering (UWE) [10,11]. Our modeling method is not a totally new one that is defined from scratch, but we compare and combine well proved design aspects of existing hypermedia design methods to improve the resulting combination with some new ideas. Fig. 1 shows the Com$^+$HDM methodological approach.

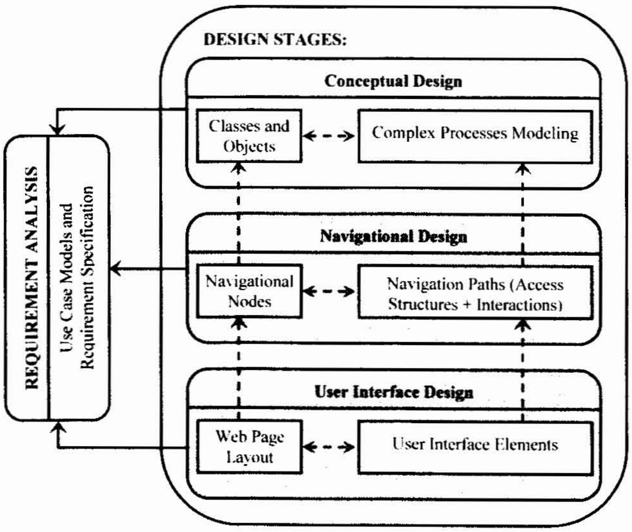

Fig. 1: Methodological Approach of Com$^+$HDM

Our method can be divided into four main stages which are performed in iterative and incremental design process. The stages are *requirement analysis*, *conceptual design*, *navigational design*, and *user interface design*. Each stage contributes to the construction of a model. In general, we use UML standard modeling elements and UML stereotypes for the applications modeling. The stereotypes are used to indicate the descriptive and restrictive properties that the modeling elements have in comparison to standard UML modeling elements. Some reviews and discussions about UML stereotypes can be found in [3]. The following sections describe the uses of UML standard modeling elements and UML stereotypes defined in our modeling approach.

3. Conceptual Design of Com$^+$HDM

Most of hypermedia design methods such as WSDM [8], OO-H [12], and UWE [11] implement user-centered approach to define clearly who will be the potential users of the application and how they behave towards the application processes. The functional requirements of the application will be easily captured according to the users' interests and navigation preferences. Results are normally presented in Use Case Models [13] and Requirement Specifications conforming to IEEE 830/1998 standard. Due to limitation space in this paper, we restrict ourselves (in our case study) to a number of functional requirements only, although many other elements should be included in the application. Fig. 2 defines some of activities performed by a student while navigating *course* webpage.

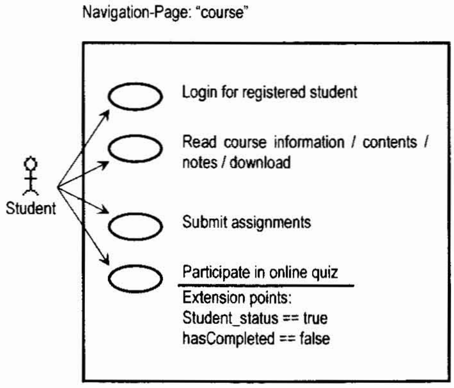

Fig. 2: Use Cases for the "Course" Navigation-Page

Navigation-page use cases diagram [2] has been implemented in our case study to capture some of complex functional requirements on particular websites in order to provide a simpler view and give better way for defining what will be navigated by the users in a particular webpage. In the *course* webpage for example, a registered user can login, read information about course, submit assignments, and participate in online quiz (which seems can be defined as complex processes). Some other results from different classified users can be modeled in a similar way according to the given examples.

3.1. Conceptual-Class Design

According to [10], conceptual design should be based on user functional requirements defined in the use cases models. The aim is to build a domain model with no consideration of navigation links, user interface, and interactive aspects. In Com+HDM, *conceptual-class design*, which is also known as *content modeling*, has to construct a UML class diagram. The main purposes are two folds; (1) to capture all classes and objects that are related in the application domain, and (2) to provide a clear view on the classes' associations. However, this design activity is less concerning on specifying complex processes.

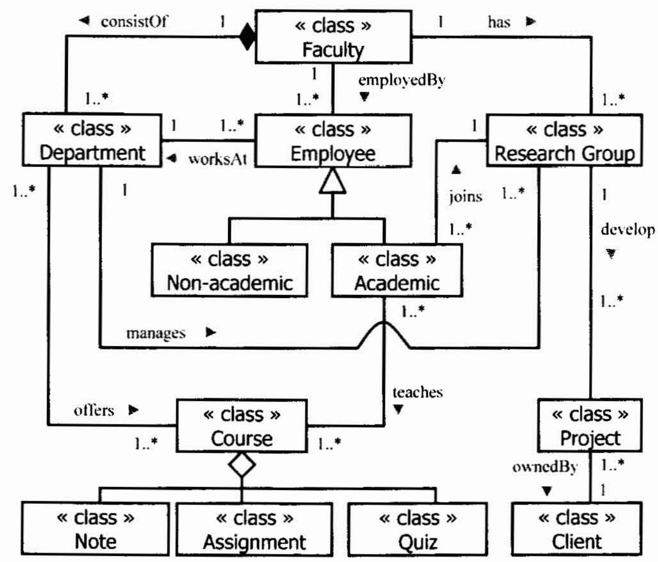

Fig. 3: Conceptual Class Model (CCM) of SLEX-Web

The result of this design activity is presented via *conceptual-class model (CCM)* as illustrated in Fig. 3. Standard modeling elements have been used for constructing the model such as class name, association, role name, multiplicity, relationship, etc. For the sake of simplicity, the CCM for the SLEX-Web hides class's attributes and operations.

3.2. Complex-processes Flow Design

Refinement of the complex processes modeling is conducted in this design activity to provide *behavior view* of the web hypermedia application, which is also called *activity / process flow design*. The main objectives are two folds; (1) to define *User Activities* – which task a user should perform (in logical and temporal order) to complete a transaction and achieve the goal, and (2) to determine both atomic (single process) and non-atomic (set of predefined processes) operations that are classified as complex processes in the application. The result is presented by UML activity diagrams (standard modeling elements), called *Complex-process Flow Model (CFM)*. In SLEX-Web case study (see Fig. 3), the focus will be centered on the class *course* which aggregated by the class *note*, *assignment*, and *quiz*. These classes will provide examples to present the existence of complex processes and how the design should be conducted.

Fig. 4 presents the main activities involved in *Quiz* class. The CFM consist a number of complex processes (non-atomic processes) which are performed in sequential and iterate fashions. At the beginning, students must validate themselves through the login process and this will require a database lookup for the particular entry. Once the students have been logged in, they could proceed to start the quiz session. Students are required to set the answer one by one and post them to the answer sheet.

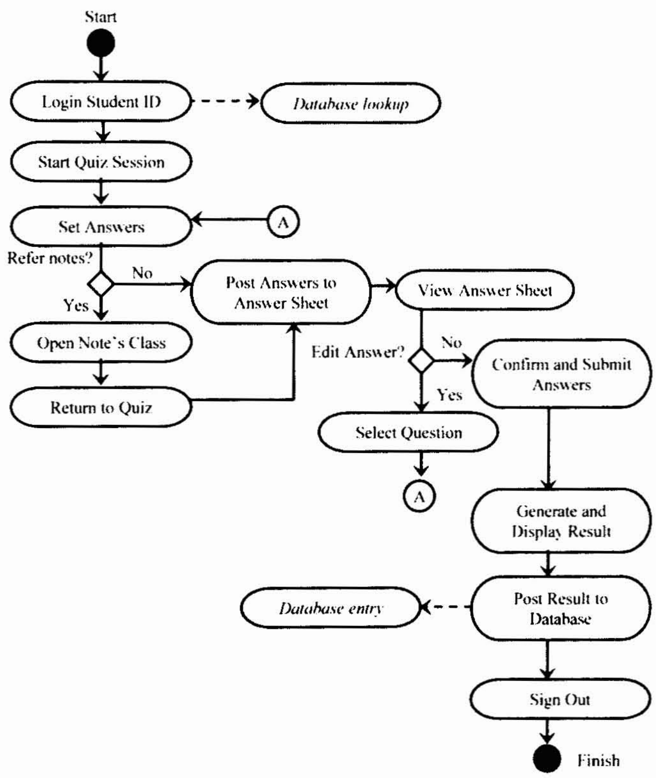

Fig. 4: Complex-process Flow Model (CFM) of class Quiz

While answering the questions, students are possible to navigate online notes (as in *note's* class), out from quiz session's page. After a while, they can resume again their quiz session. These processes repeated until all questions have been answered by viewing the answer sheet. If there are some amendments of answers, students are capable to reach the questions and change their answers. Students will confirm and submit the answers before getting the result. Result is displayed and stored in database for further record. Students may sign out after that.

4. Conceptual-process Design: Our Approach

Com+HDM design approach for modeling complex processes in web hypermedia applications is based on the two models constructed in previous sections; namely *Conceptual-class Model (CCM)* and *Complex-process Flow Model (CFM)*. In this design activity, we introduce and propose a number of new modeling elements, called *stereotypes*, improving the way of current hypermedia design method such as OOHDM [7], OO-H [12], and UWE [11] to deal with specific complex processes in web hypermedia applications. According to [18], OOHDM treats the existence of complex processes (defined as *business processes*) by partitioning the conceptual design space between *entities* and *activities (processes)* using their proposed UML stereotypes called «entity», «activity», «entity node», «activity node», and «activity node container» to indicate which partition an item belongs to. On the other hand, OO-H and UWE have proposed other solutions of dealing complex processes in web applications [16]. OO-H treats their complex processes through its *class diagram, activity diagrams,* and *mapping rules set* to transform the conceptual model into navigational model [25]. UWE however, introduces some specific classes that are part of a separate process model with clear interface to the navigational model. Two UML stereotypes were proposed by UWE, namely «process class» and «process link».

4.1. A Support for Complex Process Modeling

One of the factors that make web hypermedia applications as a complex example is the needs for them to support flow of complex processes for their evolution [17]. Ongoing research has shown that complex processes must be considered and relied on each design process. According to [18], complex processes are inadequately supported by the means of normal navigations mechanisms but they should have to integrate some process flow in order to avoid usability problems and erroneous results from being generated. Thus, this will require web designers to consider designing complex processes from the earlier design stage.

Most of hypermedia design methods have a little concern on this issue and they are incomplete or inadequate to answer this new challenge of complex processes modeling. A comparative study has been made by [21], that shows most of current hypermedia design methods are unable to address the modeling functionality of a complex process. Therefore, in the next section, we will introduce and describe our propose UML stereotypes for designing complex processes in web hypermedia applications.

4.2. The Conceptual-process Stereotypes

A brief introduction about complex hypermedia application modeling has been discussed in [1] in order to capture a wider view of critical complex processes in conceptual design phase. In the treatment of complex web hypermedia applications, we conclude and propose conceptual-process modeling elements (stereotypes) as regard to contribute on how complex processes should be systematically modeled in conceptual-process design step. The modeling elements are defined as UML stereotypes as shown in Table 1:

Table 1: The Proposed UML Stereotypes

Stereotype / Graphical Notation	Descriptions
«process class» △ Process_Class	• Will be defined as the similar way as action taken by user to perform an activity. This can be done easily by referring to the *use case* definition in functional requirement analysis. • Instance or object would be used by users during the execution of a sequence of pre-defined processes. • Inherited by two sub-classes, called *atomic class* and *non-atomic class*. • In any cases, *process class* would become the root of *non-atomic class* flow of processes.
«atomic class» ○ Atomic_Class	• Inherit the definition of *process class*. Determined by the name of action taken by user to perform an activity as being defined in *use case* definition. • Can be performed and executed independently as a single operation / process.
«non-atomic class» ⊙ NonAtomic_Class	• Inherit the definition of *process class*. Determined by the name of action taken by user to perform an activity as being defined in *use case* definition. • In contrast to *atomic class*, the execution of *non-atomic* or pre-defined processes must be performed in sequential order (they might have dependencies from each other). • *Non-atomic class* will pass information from one class to another class.
«database class» Database_Class	• Models the existence of database in application domain. • To provide a logical view of database operations between *process class (atomic class and non-atomic class)* and *database class*. • *Database class* must owned by at least one *process class*.
«process container» Process_Container	• Group and partition *process class* and all of its related modeling elements in order to indicate their relationship or dependencies. • Determine which partition an instance of processes belongs to.
——→ «Process_Link» (Stealth Arrow)	• Association between two separated classes: *conceptual class* to *process class* and vice versa. • Also known as *external link* that connects those classes.
- - - - → «Action_Link» (Dashed-Stealth Arrow)	• Association between operations taken by users in the same *process class* (*process class* to *process class*). • To force dependencies of processes and information flow in particular *process class*.
←——→ «Database_Link» (Bold Arrow)	• Association between *conceptual class* or *process class* to *database class*. • Represent the information and data operations such as

	query, lookup, entry, etc that involved with database application.

Table 1 shows the list of conceptual-process UML stereotypes used in our proposed design method. During conceptual-process design, these stereotypes are needed to cooperate with each other in order to construct the final model of conceptual design phase, called *Conceptual-process Model (CPM)*. This effort differs from other existing hypermedia design methods in a way to construct separate and yet detail class features which has been included in most hypermedia application domain.

In our SLEX-Web case study, we have considered *Conceptual Class Model (CCM)* and *Complex-process Flow Model (CFM)* in order to design the CPM. In this model, we include conceptual classes from CCM and flow of user activities from CFM to generate and implement our new stereotypes as proposed in Table 1. The CPM of our case study is shown in Fig. 5.

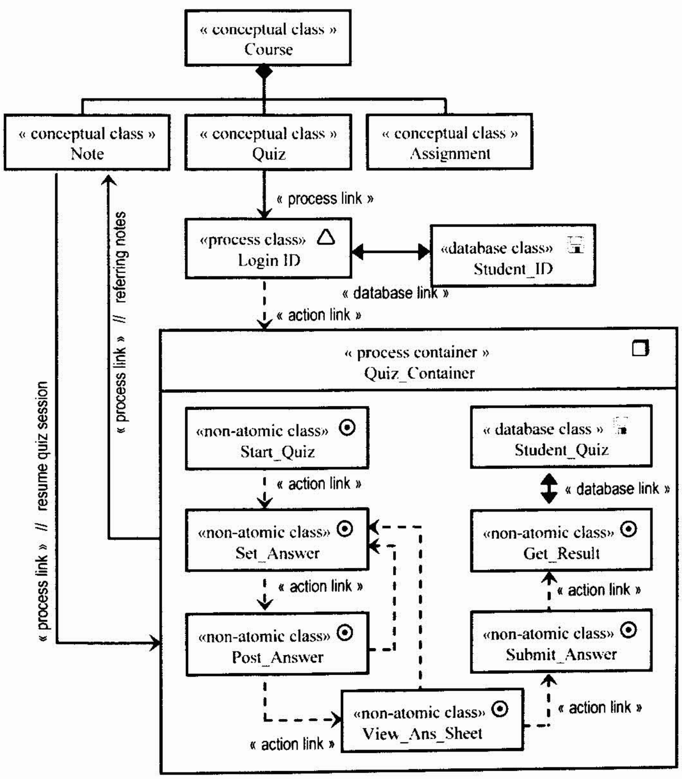

Fig. 5: Conceptual Process Model (CPM) of SLEX-Web Application

As we refer to Fig. 5, each student must hold a login ID and validate it before start a quiz session. Once validated through *Student_ID* database, student can begin their quiz session at their own pace. The quiz session is divided into a number of pre-defined processes (non-atomic processes) which defined via *non-atomic classes*. In these *non-atomic classes*, some of them might be performed in sequence or iterate fashions. All of these classes are group and partition in a container called *Quiz_Container*. However, if students are required to refer some notes while answering a question, they are possible to navigate out of the quiz session through a stereotype *process link* to reach *conceptual class note*. After a while, students can resume again their quiz session without login again. These processes proceed until students complete to answer all questions and view them through *answer sheet*. Once confirmed with all answers, students submit their final answers before get the result. All results are stored in *Student_Quiz* database.

5. Conclusions

We have presented a UML-based approach of complex web hypermedia modeling, called Com[+]HDM. Our main objective is to demonstrate a clear and systematic modeling concept of conceptual design in complex hypermedia applications. The use of UML notation has the advantage of using a well known standard and the compatibleness by many CASE tools [22]. Besides, UML also provides great facilities to present models constructed as design results based on its own systematic model-driven approach. Our focus is to concentrate on UML stereotypes as the main modeling elements in the conceptual modeling. We separate our work into three steps, called *conceptual class design, complex processes flow design,* and *conceptual process design*. Each design step produces a model to present the conceptual domain of the application, namely *conceptual class model (CCM), complex-process flow model (CFM),* and *conceptual process model (CPM),* respectively. According to [18] and [21], most of hypermedia design methods are inadequately to support modeling complex processes and we have discussed this in Section 4. Therefore as part of this work, we introduce a number of new modeling elements (stereotypes) to model a flow of complex processes in hypermedia application. Each UML stereotype has their own definition and reacts as a separate modeling class to represent the classification of complex process. These stereotypes can be summarized as a *conceptual process pattern* as shown in Fig. 6.

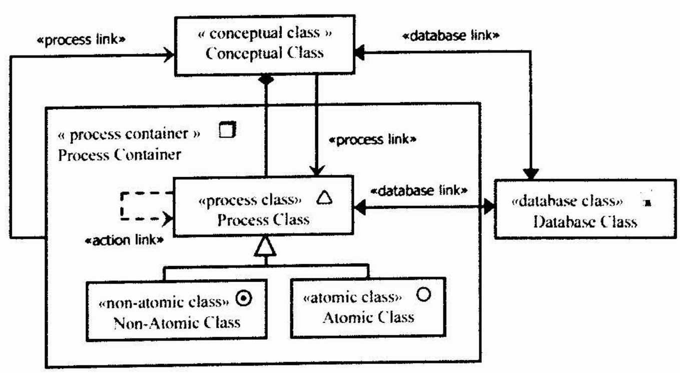

Fig. 6: The definition (pattern) of UML stereotypes for Conceptual-Process Design in Com+HDM

To conclude, we summarize and propose a *conceptual process pattern* for complex hypermedia design in Com+HDM as shown in Fig 6. This conceptual process pattern can be referred as the core modeling elements used in the *conceptual-process design* step.

6. References

[1] A.H. Jantan, P. Sumari, and S. Sulaiman, "Design Processes for Web-based Hypermedia Engineering" *The 3rd Malaysian Software Engineering Conference '07*, Selangor, pp. 182-186, Dec 3-4 2007.

[2] A.H. Jantan, P. Sumari, and S. Sulaiman, "Systematic Design Processes of Hypermedia Engineering: A Case Study of SpLEA-Web", *1st Regional Conference on Computational Science and Technologies '07*, Sabah, pp. 154-158, Nov 29-30 2007.

[3] A.H. Jantan, P. Sumari, and S. Sulaiman, "UML-based Hypermedia Design Method: A Review on Design Processes and Modeling Elements", *Computer Science Postgraduate Colloqium (CSPC'07)*, Universiti Sains Malaysia, June 2007.

[4] M. Lang and B. Fitzgerald, "Hypermedia Systems Development Practices: A Survey", *IEEE Software*, Volume 22, Issue 2, pp. 68-75, March-April 2005.

[5] Z.S. Qian, H.K Miao, and H. Tao, "An Approach to Modeling Hypermedia Web Applications", *The 6th International Conference on Grid and Cooperative Computing 2007*, pp. 847-854, 16-18 August 2007.

[6] T. Isakowitz, A.S. Edward, and P. Balasubramaniam, "RMM: A Methodology for Structured Hypermedia Design", *Communication of the ACM*, Volume 38, Issue 8, pp. 34-44, 1995.

[7] D. Schwabe and G. Rossi, "Developing Hypermedia Application using OOHDM", *Proceedings of Workshop on Hypermedia Development Process, Methods, and Model*, 1998.

[8] De Troyer, "Audience Web Design", *In Information Modeling in the New Millenium, Eds Matt Rossi and Keng Siau*, IDEA GroupPublishing, 2001.

[9] De Troyer and C.J. Leune, "WSDM: A User-centered Design Method for Websites", *Proceedings of the 7th International WWW Conference on Computer Networks and ISDN Systems"*, Elsevier, pp. 85-94, 1998.

[10] N. Koch, A. Kraus, and R. Hennicker, "The Authoring Process of the UML-based Web Engineering Approach", *In Daniel Schwabe, 1st International Workshop on Web-oriented Software Technology (IWWOST01)*, online publication, June 2001.

[11] N. Koch and A. Kraus, "The Expressive Power of UML-based Web Engineering", *2nd International Workshop on Web-oriented Software Technology (IWWOST02)*, D. Schwabe, O. Pastor, G. Rossi, and L. Olsina, editors, CYTED, pp. 105-119, June 2002.

[12] J. Gomez and C. Cachero, "OO-H Method: Extending UML to Model Web Interfaces", *Information Modeling for Internet Applications*, pp. 144-173, 2003.

[13] G. Booch, I. Jacobson, and J. Rumbaugh, "The Unified Modeling Language User Guide", *The Addison-Wesley Object Technology Series*, 1998.

[14] A. Ginige and S. Murugesan, "Web Engineering: An Introduction", *IEEE Multimedia*, Volume 8, No. 1, pp. 14-18, Jan-March 2001.

[15] P. Avgeriou and S. Retalis, "CRITON: A Hypermedia Design Tool", *Multimedia Tools and Applications*, 27, 5-21 Springer, 2005.

[16] N. Koch, A. Kraus, C. Cachero, and S. Melia, "Modeling Web Business Processes with OO-H and UWE", *Proceedings of the 3rd International Workshop on Web Oriented Software Technology (IWWOST)*, July 2003.

[17] D. Distante, G. Rossi, G. Canfora, and S. Tilley, "A Comprehensive Design Model for Integrating Business Processes in Web Applications", *International Journal Web Engineering and Technology*, Volume 3, No. 1, pp. 43-72, 2007.

[18] H.A. Schmid and G. Rossi, "Modeling and Designing Processes in E-Commerce Applications", *IEEE Internet Computing*, Volume 8 Issue 1, pp. 19-27, Jan-Feb 2004.

[19] J. Dietz and H. Shouten, "Modeling Business Processes for Web-based Information Systems Development", *Proceeding of the 1st International Conference on Web Information System Engineering*, Volume 1, pp. 270-282, June 2000.

[20] O. De Troyer and S. Casteleyn, "Modeling Complex Processes for Web Applications using WSDM", *Proceedings of the 3rd International Workshop on Web Oriented Software Technology (IWWOST)*, July 2003.

[21] S.S. Selmi, N. Kraiem, and H.B. Ghezala, "Toward a Comprehension View of Web Engineering", *International Conference on Web Engineering (ICWE05)*, Sydney, Australia, pp. 19-29, July 2005.

[22] R. Hennicker and N. Koch, "A UML-based Methodology for Hypermedia Design", *UML'2000 – The Unified Modeling Language*, Advancing the Standard, LNCS 1939, York, England ©Springer Verlag, October 2000.

[23] S. Ceri, P. Fraternali, and M. Matera, "Conceptual Modeling of Data-Intensive Web Applications", *IEEE Internet Computing*, Volume 6, Issue 4, pp. 20-30, July – August 2002.

[24] M. Brambilla, S. Comai, and C. Tziviskou, "Exception Management Within Web Applications Implementing Business Processes", *C. Dony et al. (Eds): Exception Handling, LNCS 4119*, Springer-Verlag Berlin Heidelberg, pp. 101-120, 2006.

[25] J. Gomez, C. Cachero, and O. Pastor, "Extending a Conceptual Modelling Approach to Web Application Design", *Conference on Advanced Information System Engineering*, 2001.

[26] L. Baresi, F. Garzotto, and P. Paolini, "From Websites to Web Applications: New Issues for Conceptual Modeling", *ER'2000 Workshop on Conceptual Modeling and the Web*, LNCS 1921. Springer-Verlag, pp. 89-100, 2000.

[27] D. Distante, G. Rossi, and G. Canfora, "Modeling business processes in web aplications: an analysis framework", *Proceedings of the 2007 ACM Symposium on Applied Computing*, pp. 1677-1682, 11-15 March 2007.

A Study on Landmark Identification for Craniofacial Data Registration

Pan Zheng[1], Rozniza Zaharudin[1], Bahari Belaton[1], Zainul Ahmad Rajion[2]
School of Computer Sciences[1], School of Dental Sciences[2]
Universiti Sains Malaysia
panzheng@gmail.com, drubanks99@yahoo.com, bahari@cs.usm.my, zainul@kck.usm.my

Abstract

The focus of this study is to describe the development of a method identifying landmarks on an isosurface visualized from CT generated craniofacial data in a 3D environment. The work is achieved by using graphical subprogram library, Visualization Toolkit. This development of this study facilitates current craniofacial surgeries. A 3D-based automatic landmark detection method is proposed at the end.

1. Introduction

Craniofacial surgeons rely on landmarks on synthetic skull model to analyze and plan their operations conventionally. Landmarks are obtained through direct measurement on 2D images from X-ray, or ultra sound images in many occasions. The identification is a tedious and time consuming task which often depends on ones' previous experiences and understanding with certain level of arbitrary. As computer-assisted surgery is becoming more and more popular because of safety, accuracy and less side effect and damage [1], by taking the advantage of current widely used computed tomography (CT) and Magnetic Resonance Imaging (MRI) technologies, and using proper techniques and tools to identify landmarks directly on digitized craniofacial data, this fills the gap. Craniofacial visualization as a prerequisite is a process of creating interactive visual 3D representation of human skull. It tremendously facilitates the process. Landmarks are placed in a visualized environment. Here in this study, landmarks are defined as salient points which denote distinctive feature on hard tissue of human head and have strong anatomical relationship among them. Registration of a deformed skull to a standard model can be conducted by various ways [2]. Landmark based registration (LBR) is a common and noninvasive practice in this domain.

The aim of this study is to describe the development of an efficient and convenient method for identifying craniofacial landmarks on CT generated data using Visualization Tool Kit (VTK) by Kitware, Inc.

2. Related works

Craniofacial Landmark identification research is intensively conducted by experts from both computer science and medical science fields. Our researchers have previously conducted landmark placement analysis based on anatomical regions. The approach uses 3D CAD files for visualization and landmark placements [3]. This approach is limited by CAD environment and lack of flexibility of further development. Grau et al [4] proposed a method on landmark identification on 2D X-ray images. This 2D implementation is consisted of two phases which are line detection model and point detection model. Line detection uses zero crossings detection of the Gaussian Laplacian (figure 1) and for point detection, it uses mathematical morphology techniques. This method provides a very convincing landmark detection approaching, but it suffers from high computation cost.

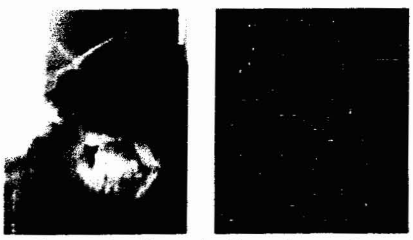

Figure 1. **Grau's line detection**

A Complicated neural network approach uses machine learning is given by Feghi [5]. It concerns on pattern, which requires a predefined contour map. Such approach requires high quality of images and hardly to implement where there is a need for additional landmarks.

3. Landmark identification

Our study is carried in 3D environment. It includes five related phases, which are data acquisition, visualization, data preprocessing, reference model construction and application of landmark placement.

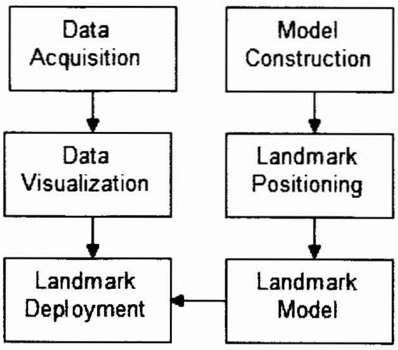

Figure 2. **Basic flow of the design**

3.1. Data Acquisition

The sample data sets used in this research are captured with a GE Lightspeed Plus CT Scanner System at the Department of Radiology, Hospital Universiti Sains Malaysia. The scans are conducted in axial manner. There are often 160 to 210 slices for a human head. The data produced by CT scanner are sent to GE Advantage workstation.

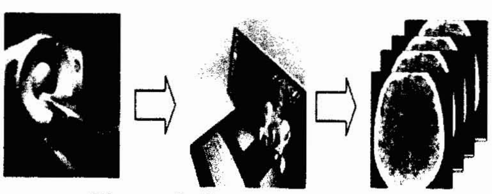

Figure 3. **Data acquisition**

A GE Advantage workstation is a CT data analysis and repository system which is based on Sun's Solaris operation system. The CT data are saved in DICOM (Digital Imaging and Communications in Medicine) format.

3.2. Visualization

There are many visualization tools available. Visualization Took Kit (VTK) from Kitware is powerful, open source subprogram library which is built in C++. Besides C++, it offers many wrapping programming languages such as JAVA, Python and Tcl/tk [6].

From data visualization perspective, there are generally two methods which are iso-surface rendering and volume rendering. Iso-surface construction is using Marching Cubes. The visualization pipeline is illustrated in figure 4.

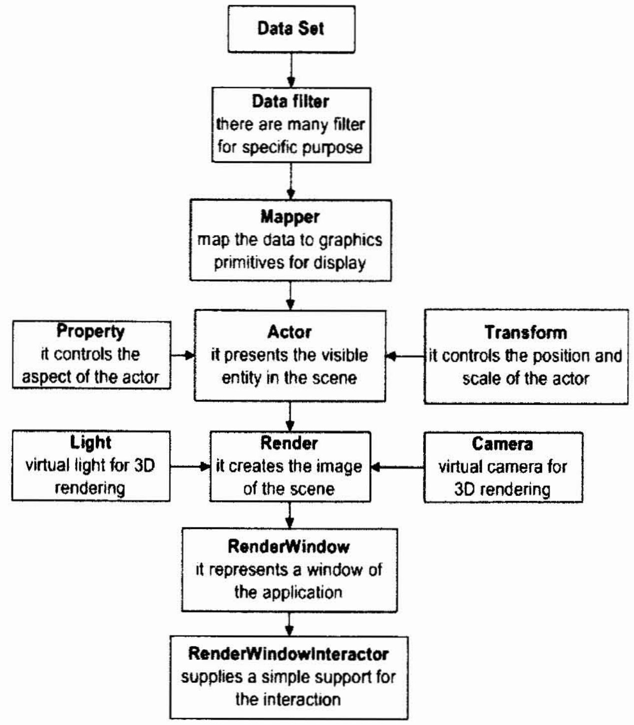

Figure 4. **Visualization pipeline**

3.3. Data preprocessing

The actual DICOM data is fairly huge, which is inefficient for data loading and manipulation. Data preprocess reduces the data size by extracting the iso-surface of interest and sub sampling. The result is desirably small, yet aliasing.

We apply Gaussian Smooth to eliminate jaggies. The final data is saved in .vtk native data format.

3.4. Reference model construction

Model construction refers the process that builds some kind of perfect "average" human skull. There are two methods of dealing this issue. One idea is to apply cephalometric along with some computer graphics tools to get the reference model. Another approach is based on a database of human skull images. The collaboration with Hospital Universiti Sains Malaysia (HUSM) provides us relevantly enough data set to form reference model construction.

Due to the nature of the scan, not all skulls are in the same orientation. Osteometric Scaling is applied to construct a 3D Cartesian coordinate system to fit sample data. Coordinate system integration is done by our researchers by fitting the digitized craniofacial data into a standard position named Frankford Horizontal Plan (figure 5) [7, 8].

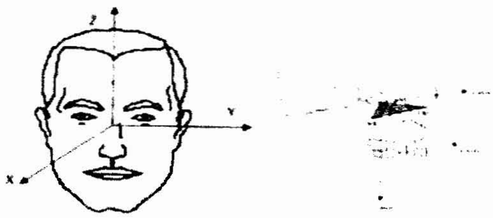

Figure 5. **The coordinate system of Frankfort Horizontal Plane**

3.5. Landmark placement

A test application is created using Tcl/tk with functions of "create a landmark", "delete a landmark", "save landmark set", "load craniofacial data set" and etc.

Here briefly describes how to create a new landmark:

- A point is created under mouse point with the coordinate (px, py, pz).
- From 2D view, set px and py with current mouse location.
- Check whether a data cell is selected under current location on 2D screen. (On surface?)
- If no, quit. Else update the point coordinates to 3D coordinate values in the visual 3D environment.
- Add the point to landmark set list and process the pipeline to render actor (a small sphere) on the skull surface.

Another function is called in order to delete a landmark. This is done by pointing out the point from the landmark set list, and removes the corresponding small sphere actor.

These landmarks set are then saved into a .vtk format, as saved-landmark cloud. The next time when user wishes to view the skull with landmark placements, users can just load the landmarks based on the saved-landmark cloud earlier.

Hereby landmark clouds placed on a reference model is called landmark model. Landmark model can be placed as reference to surgeons, when a sample skull is loaded in the application with integrated coordinate system. Landmark model might not exactly suit the subject's head. A manual adjustment is required necessarily.

4. The Landmarks Reference Model

Based on Feghi's paper [5], this study chooses a set of landmarks for the purpose of the reference model. These landmarks are selected and organized into a table based on the landmark clouds saved with our application, with the respective x-y-z coordinates.

Shown in Figure 6 is the snapshot of the landmark coordinates table with three subjects from the database.

		PATIENT 1			PATIENT 3			PATIENT 5	
	X	Y	Z	X	Y	Z	X	Y	Z
Bregma	121.537	183.626	29.396	125.161	131.876	18.9361	127.956	179.421	18.4867
Nasion	122.875	213.204	137.978	120.213	198.499	116.693	127.735	215.174	126.469
Sella	62.0225	171.503	144.314	65.717	158.792	139.717	69.2061	176.665	140.969
Subnasal	124.449	191.9	191.471	121.475	202.891	168.153	125.951	198.856	179.936
Subspindale	120.028	191.856	185.001	117.873	200.222	161.99	121.847	199.321	177.134
Pognion	126.042	160.228	236.59	121.364	191.626	230.588	124.612	174.371	222.32
Gnathion	126.478	153.97	239.157	121.192	189.316	234.56	125.491	165.202	223.972
Manton	126.081	149.718	236.503	122.855	182.757	234.954	125.451	160.044	221.448
Gonion	76.7261	105.497	183.425	77.7674	119.218	206.486	76.7467	120.924	183.346
Articular	73.4101	119.016	149.747	74.4732	112.527	160.157	71.7811	125.196	138.78
Condylion	66.1994	129.326	140.554	66.1324	122.391	149.244	64.7058	135.28	134.211
Orbitale - Down	93.9769	195.062	158.945	91.0322	186.223	141.232	95.4695	199.917	152.977
Orbitale - Up	93.4842	212.308	124.834	88.9245	192.193	105.376	95.6744	216.62	116.687
Orbitale - Left	110.474	204.489	137.377	109.538	189.485	117.284	115.297	206.916	130.006
Orbitale - Right	76.3534	190.293	137.064	75.8977	176.512	125.032	79.4177	197.947	131.37
Anterior Nasal Spine	105.85	195.043	168.277	103.094	190.202	145.792	101.905	197.846	161.091
Posterior Nasal Spine	71.5995	166.477	171.836	78.0903	172.181	164.526	81.777	179.093	168.226
Anterial for Occlusal	93.1569	149.922	190.929	95.0293	156.564	188.088	94.5372	158.795	185.17
Posterior for Occlusal	104.48	178.167	207.815	99.3583	190.882	192.853	106.367	187.591	197.482
Basion	63.9052	100.263	150.859	72.0091	92.6613	171.572	84.3133	98.4761	139.694

Figure 6. **Snapshot of landmark coordinates table**

In Figure 7 below shows a statistic analysis on the landmarks' coordinates integrated with the patient's landmarks. This snapshot is a portion of the full number of patients.

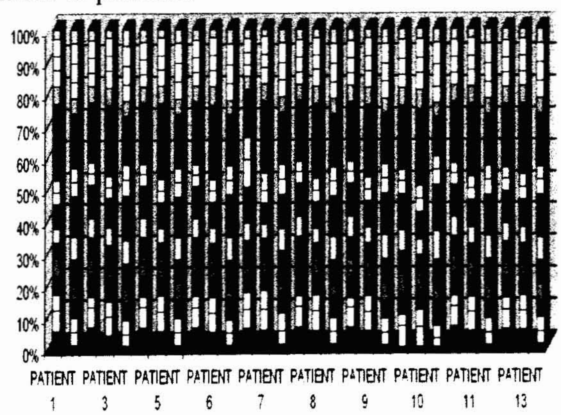

Figure 7. **Statistic analysis on subjects' landmarks**

5. Example and Analysis

For the research trial, several sets of DICOM data are loaded to our application.

Figure 8 shows the visualization result of the digitized skull dataset, before the identification of landmarks is performed. The results are then shown in Figure 9 where landmarks are plotted based on a certain table model (shown in greens). Note that in this example, the Gaussian smooth technique was not applied.

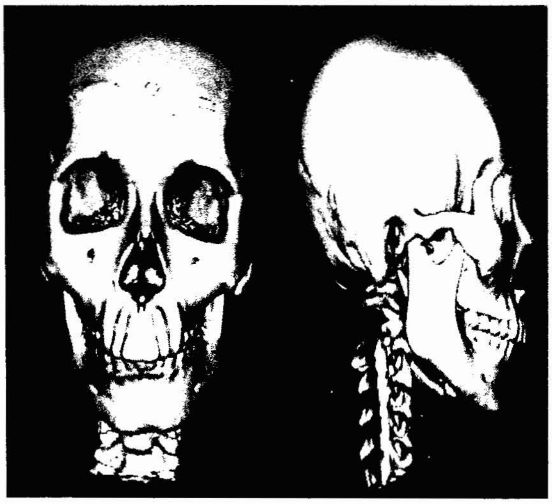

Figure 8. **Visualization of digitized skull**

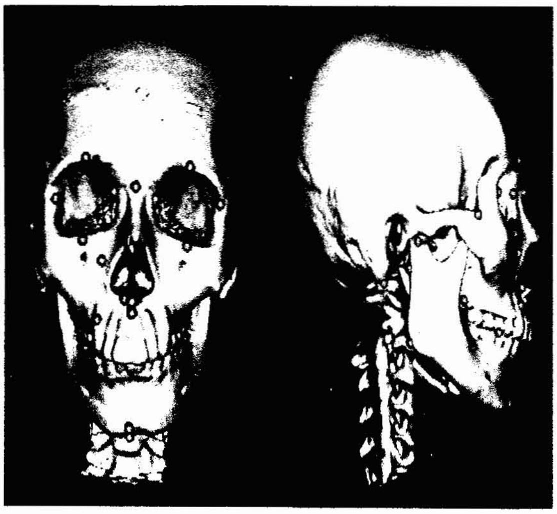

Figure 9. **Landmark placement on subject's skull**

Now, looking at another different sample dataset, Gaussian smooth technique is applied. Figure 10 shows the visualization result of the dataset before the identification of landmarks was performed. The results are then shown in Figure 11.

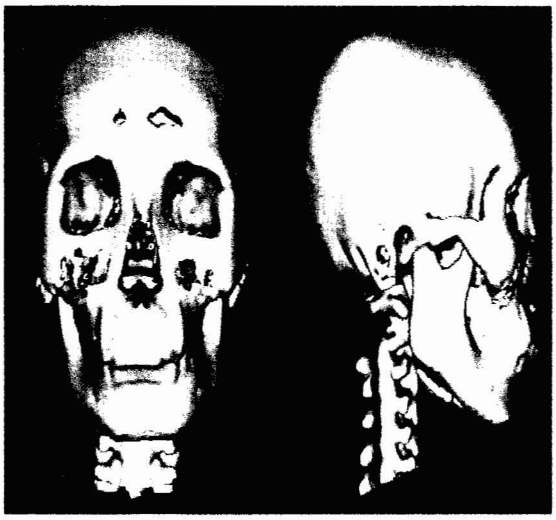

Figure 10. **Result Screenshots from front and side view (before plotting landmarks)**

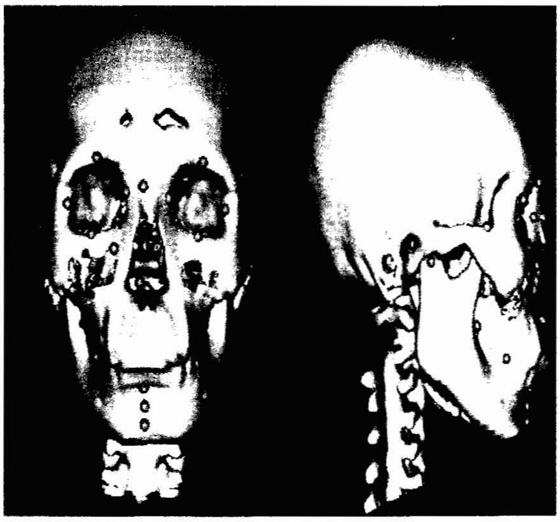

Figure 11. **Result Screenshots from front and side view (after plotting landmarks)**

From the same skull dataset, we can conclude that the landmarks identification can be seen clearly in Figure 7 and Figure 9. Since we wanted the results of smooth and rough surfaces, the differences in these two results can easily be compared. Although these both are derived from the same patient's dataset, the differences of result are apparently obvious.

6. Future works

In this study, we look into how craniofacial visualization and landmark identification is achieved by using Visualization Toolkit (VTK) and wrapper language Tcl/tk. Also we manage to create program to identify and manipulate landmarks on the hard tissue of the digitized craniofacial data. The result is as expected.

In future, the landmark identification can be conducted in an automatic way. We have looked at the surface construction techniques, such as Jules Bloomenthal's polygonization of implicit surface [9]. By studying the surface forming algorithm, we try to extract the feature lines on the iso-surface of craniofacial data. Then landmark can be identified easily alone the feature lines.

References

[1] Michelle Meadows, "Computer-Assisted Surgery: An Update", *FDA Consumer Magazine*, U.S. Food and Drug Administration, Maryland, Volume 39, Number 4, July-August 2005.

[2] Lisa G. Brown, "A survey of image registration techniques" *ACM Computing Surveys* (CSUR) archive Volume 24, Issue 4, 1992.

[3] Wan Abdul Rahman Wan Harun, Zainul Ahmad Rajion, et al "3D CT Imaging for Craniofacial Analysis Based on Anatomical Regions" *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference* Shanghai, China, September 1-4, 2005

[4] V. Grau, M. Alcaniz, M. C. Juan, et al, "Automatic Localization of Cephalometric Landmarks" *Journal of Biomedical Informatics*,Volume 34, Issue 3, 2001, Pages 146-156.

[5] I. El-Feghi, M. A. Sid-Ahmed and M. Ahmadi, "Automatic localization of craniofacial landmarks for assisted cephalometry" *Pattern Recognition*, Volume 37, Issue 3, 2004, Pages 609-621.

[6] William J. Schroeder, VTK an Open-Source Visualization Toolkit, Kitware, Inc. 2001

[7] Zainul Rajion, Deni Suwardhi, et al "Coordinate Systems Integration for Development of Malaysian Craniofacial Database" *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference* Shanghai, China, September 1-4, 2005

[8] Farkas, L.G. ed. *Anthropometry of Head and Face. 2nd ed.* New. York: Raven Press. 1994

[9] J. Bloomenthal, ''Polygonization of Implicit Surfaces,'' *Computer Aided Geometric Design* Vol. 5 1988, pp. 341-355.