

Model-Based HCI System Development Methodology

Kyung Won Cha and Changbeom Choi^(✉)

Handong Global University, Handong-ro,
Heunghae-eup, Kyungbuk 37554, Republic of Korea
{2120751,cbchoi}@handong.edu
<http://www.handong.edu/eng>

Abstract. Nowadays, many Human Computer Interaction(HCI) systems are developed and utilized to the various domains, such as medical field, and entertainment area. To develop an HCI system, a developer should consider developing hardware elements and software elements at the same time. One of the obstacles obstructing the HCI system is that, in the absence of hardware, there is no development environment. This paper proposes a model - based HCI system development environment and development methodology using it. The proposed development environment focuses on the Natural User Interface (NUI) which utilizes HCI's human movement, and it can replace human and sensor devices by using basic unit model. Also, the developer can test and develop the HCI system by creating a new model by synthesizing the basic unit model to build various patterns of the environment.

Keywords: Development methodology · Simulation-based development · Simulation-in-loop HCI system

1 Introduction

As the technology develops and spreads widely, the computer system has spread rapidly in everyday life. Most of the devices, such as smart TV, smart home, as well as smartphones, need interaction between human and computer system and concept of HCI has become necessary [1]. Interestingly, the entertainment domain researched the HCI systems and most well known HCI systems are game consoles. Such HCI systems can be categorized by two categories, controller-less system and system with controllers. The Nintendo Wii [2] and Oculus RIFT [3] uses separate controllers to sense body movements and recognize them as gestures, while Microsoft Xbox Kinect [4], using depth and vision sensors, users do not need a separate controller. Nevertheless, the workflows of those HCI systems are the same. Following Fig. 1 shows the workflows of the HCI system.

As shown in the figure, a sensor receives the person's motion. When a sensor senses the motion, the hardware will handle the motion and generate a signal. Then the hardware will send a sequence of the signal to the software, and the

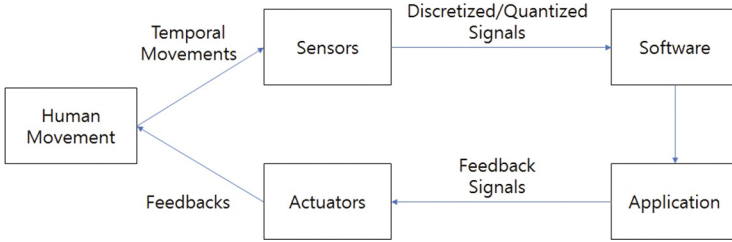


Fig. 1. Workflow of HCI systems

software will process it as an event. After recognizing an event, the software will notice the application, which developer implements the business logic. Then the application will handle the event, and it will generate proper feedback to the human via actuators. Therefore, a developer should consider the hardware, software, and human motion inputs to implement HCI system. The *hardware* is a sensor system that detects the human motions. The *software* handles signals from the hardware and generates an event. Also, the application can be considered as software, but it implements the business logics, such as assessment logic to give points to users. The *human motions* is the most delicate part during the development of the HCI system. Since a human movement may differ by age, gender, and physical size.

By considering considerations of developing an HCI system, this paper develops the development environment of HCI systems. In particular, this paper considers developing NUI system and its development methodology, since device based HCI system has common workflows. Also, most of the motion recognitions are based on the position of the hand controller; the research provides various support method. The supporting methods are depth image handling and skeleton handling. Since an HCI system focuses on detecting human behavior and generating feedback using the actuator, hardware and software should be developed first for the HCI system construction, and then the integrated system should be built by the developer in the existing development environment. That is, during the HCI system development process, the developer must develop and test the software part after the system hardware part is completed, which limits the development environment. Also, since the testing an HCI system is a Human-In-The-Loop (HITL) test in which a human is required to generate an input, even if a problem is found during testing, the problem can not be reproduced exactly every time. Therefore, this study proposes an environment to make HCI system development more efficient and proposes a methodology to utilize it.

2 Related Work

The existing HCI-related development methodology includes the Human-Centered Systems Development Life Cycle, HCS DLC [5,6]. The Fig. 2 shows the cycle. It is developed in the Systems Development Life Cycle, SDLC [7],

which divides the system development into manageable steps. The methodology aims to satisfy both the organization and human needs in the overall system development process. In order to develop human-centered systems, developers have to consider the principles and guidelines for the development of a system that encompasses as many users as possible, integrates knowledge of various fields, and encourages many interactions. The characteristic of HCSDLC is that the analysis part includes the user acceptance test, which is one of the tests, and it can be seen that the HCI component is continuously considered throughout the process.

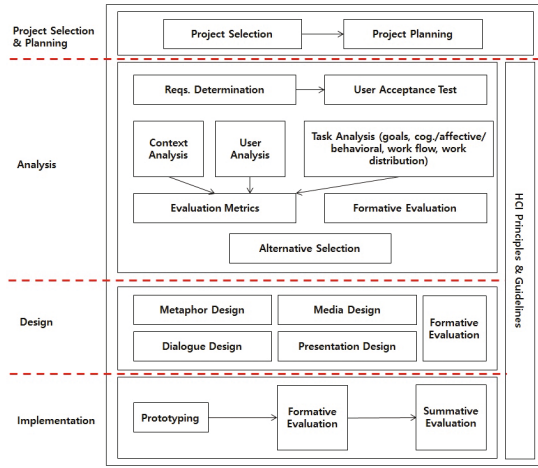


Fig. 2. Human-centered systems development life cycle

In the HCI system development, testing is an indispensable factor. Due to the nature of the HCI system, it must receive input from a sensor or controller from the outside and respond to it, but there is no specific reference to this in the existing methodology. Although there is a limitation in the development of the software in the future when there is no hardware in the past, this study proposes a methodology to regenerate and combine new input through several existing input data in order to provide an environment that can simultaneously develop software and hardware even if there is no hardware such as sensors required for the system.

3 HCI System Development Methodology

Aforementioned, the hardware in the HCI system receives a human’s motion. The hardware system has various sensors to detect human motions, such as Infrared camera, multiple camera system and microphones. Since such sensors handle continuous signals, the software part of the HCI system should process

continuous signal to generate events. Also, the HCI system has business logics as software. As a result, the development methodology of the HCI system consider integrating independent systems, such as hardware system and software system. Moreover, the developer may consider developing the system when the system does not exists. Unlike the existing development methodology, the development of the HCI system should consider hardware and software at the same time. It is also difficult to control the test environment of the HCI system because human motion is in the flow of the system. If a tester finds an error during a partial implementation and test, the tester must repeatedly generate the same operation at the same rate to find out what the error is. However, humans are not always able to generate the same motion, so human may not be sufficient for the development process of the system, and the results may be inaccurate and inefficient. Therefore, to help developer and tester of an HCI system, this study proposes a simulation based workflow of HCI system as shown in Fig. 3.

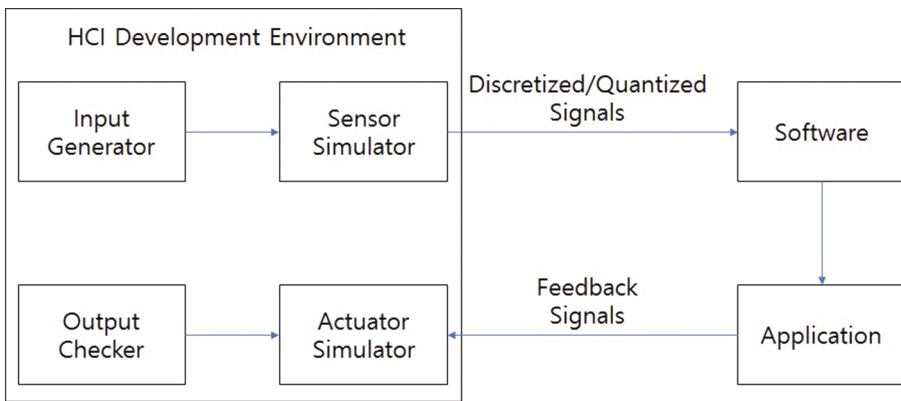


Fig. 3. Proposed simulation based workflow

A sensor simulator can replace human motion and sensor parts in the typical HCI system and testing flow by defining a model representing the main body movement. This input is used to process in software and applications to check if the desired feedback results. By using the test environment, developers can utilize some of the existing resources and use it to regenerate various body movements according to the needs and test them.

The steps for implementing the proposed testing environment are divided into three stages as shown in Fig. 4: preparation, development, and integration.

In the first preparation step, it is determined which data to prepare for a certain target movement, and information about depth and skeleton is extracted and stored in files. The preparation phase can be created using the sensor for development or data to be used, or it can be replaced with existing data used in other projects. In the subsequent development stage, development based on testing is performed using the data created in the preparation stage. It decides

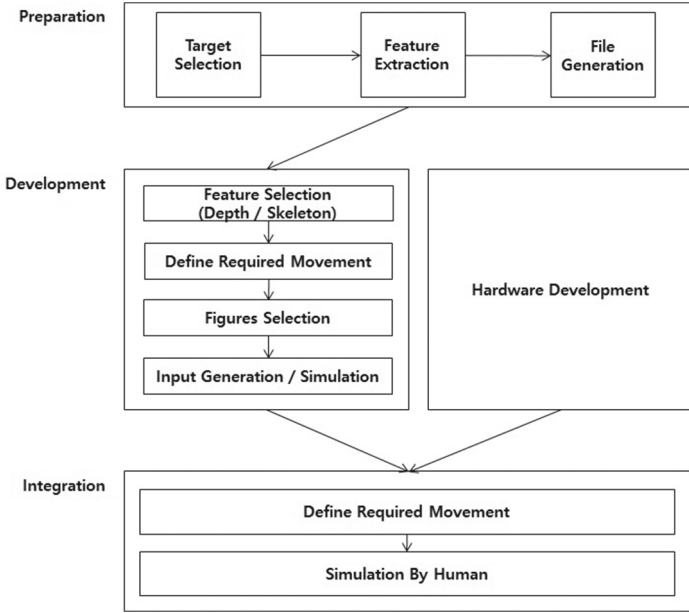


Fig. 4. Proposed HCI development methodology based on simulation

what kind of data it needs and defines the required movements for it. If developers select the depth data, select the shape of the hand after the motion definition, and then create the input and test the HCI system. The HCI system itself requires both humans and software and hardware, so if the hardware is not available, or has not been developed yet, or is lacking, further development can be difficult. It provides an environment that can be developed using the data prepared in the preparation stage, and the software can be unaffected by the speed of hardware development, and the hardware can be fully developed. Also, developers can perform the testing by reducing the other variables in that the movements can be continuously and exactly created on the same stage. By using this, input data for frequently occurring errors can be made; it helps to make testing easy. In the Integration step, developers define the required motion and test it. In the development stage, simulation is performed using existing data, but at integration stage, the actual person enters the testing loop and performs testing using the actual hardware and software. During the development process, the development and integration stages are continuously repeated, and this study proposes a testing-based development method based on the simulators.

To help a developer and tester this study provides event synthesis concepts. The Fig. 5 shows the simplified concept of motion generation.

As shown in the figure, Model₁ and Model₂ denotes the events of the human movements. Each model contains event scenario so that the model generate timed event sequence which denotes timed hand positions. In other words, the




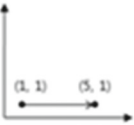
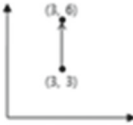
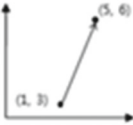
	Model ₁	Model ₂	Synthesized Gesture
Coordinate Transition	$(x_1, y_1) \rightarrow (x_1', y_1)$	$(x_2, y_2) \rightarrow (x_2, y_2')$	$(x_1, y_2) \rightarrow (x_1', y_2')$
Event Direction			
Coordinate Value			

Fig. 5. Event synthesis

Model₁ generate timed events from position (x_1, y_1) to (x_1, y_1) . Also, the Model₂ generate timed events from position (x_2, y_2) to (x_2, y_2) . When a developer wants to test pattern starts from (x_1, y_2) to (x_1, y_1) , the developer should acquire test data from human. Therefore, this study proposes event synthesis concept. By using the first input event, moving from left to right to (x_1, y_1) to (x_1, y_1) , and the second input event, moving from bottom to top moving from (x_2, y_2) to (x_2, y_2) , it may create new input event data, which start from left bottom, (x_1, y_2) , to right top, (x_1, y_2) . The new input event data uses the position and amount of change of the x coordinate from the first input event data, and the new input event data is generated as follows using the position and the change amount of the y coordinate from the second input event data. By applying the method, a developer and a test may generate multiple motions using simple input event data, and conversely, one motion can be separated into multiple motion data.

4 Implementation Using Xbox Kinect

To show the feasibility of the proposed methodology, this research implements a HCI development environment using Xbox Kinect. Since Microsoft provides software interface to fetch event data from Xbox Kinect, we have utilize the Kinect to collect motion data and implemented a simulator to generate timed events. Also, the environment has event synthesis modules that synthesize the various events.

4.1 Preparation

This study uses Microsoft Xbox Kinect to determine the motion that wants to extract and extract the motion of the base unit. All data is recorded in raw data for later usage by the user and prepare a shaping unit that can give the actual movement shape to the motion created by the base unit input. In other words, if the generated file is depth information, it will be the depth raw data, the center point of motion, and the hand shape file. In a case of the skeleton, it is a file corresponding to the coordinates of each position of the skeleton.

4.2 Input Data

This paper focuses on the development of NUI system and uses Kinect composed of RGB sensor, depth sensor and IR sensor to construct environment of research and sensor data based model. In this paper, depth data and skeleton data which are useful for three-dimensional position and motion recognition are studied. In case of depth data, input file is defined as three. The depth data is binary data of the depth information coming at a rate of about 30 fps and the other is text data including the coordinates of the center point of the recognized object, hand. Finally, the shape data is hand shape data to be added to the motion to be generated later. Binary depth data can help developers to develop when there is no hardware. Text files help developers understand the movement of data and can be used to reconstruct data to predict and change the user's movements. In addition, developers can add one more input, depending on needs, which is the data that helps shape the shape already prepared. In this case, when it is necessary to grasp the movement of a specific shape, it is used to regenerate a desired type of input data. For skeleton data, input file is defined as one. This is the coordinate of each skeleton joint and can be used to track the movement of each segment.

4.3 Input Generation

There are three ways to create a new motion: modifying the coordinates in the text file, separating and combining existing data, and creating the desired motion with the simulator. In this paper, motion generation from depth data, motion generation from skeleton data, and motion generation from simulator are discussed.

Depth. This study proposes a method to generate new motion using information of motion of existing depth data. The existing input data consists of a binary file, which is raw data, and a text file, which is a coordinate of the center point. There are several steps to export information about the movement of 'hand' which is meaningful data. Handle only the depth data within a certain depth to grasp the motion of the hand without accepting the movements other than the hand. In addition, the distance of the image is transformed, and the focal point

is recorded in the file as the center point of the hand that find the furthest point from the outside.

The raw data of the depth information is read from the created file through the following procedure. If the depth is within a certain range for understanding the movement, the RGB value is made 0 and the hand shape of the part is made white. The input₁ and input₂ images above in Fig. 6 show the hand shape created from the raw data. input₁ is part of the image moving from left to right, and input₂ is part of the image moving from bottom to top. In addition to restoring the two inputs, when generating new motion data, information about each motion is needed, and information corresponding to each motion is recorded in a text file for each input data. The third image in Fig. 6 shows the center point for the new movement from left bottom to right top using the coordinates from left to right and from bottom to top of input₁ and input₂. Tracking the movement can be done as follows. If developers have the shape of the hand needed, developers can use it to regenerate the new movement based on its center point.

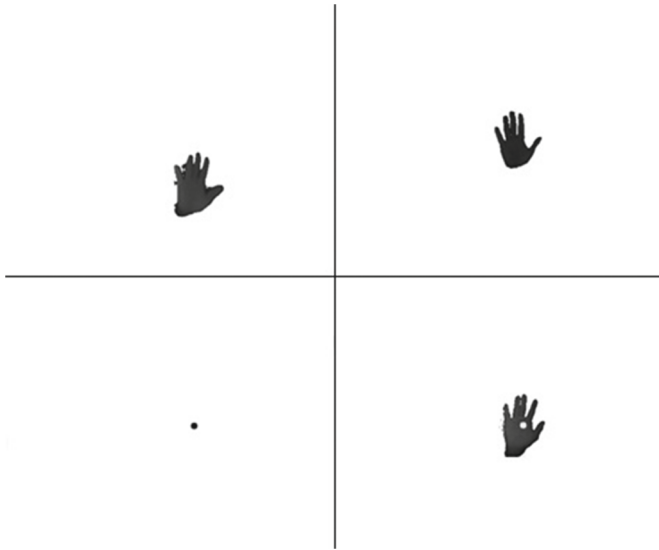


Fig. 6. Left-top is input₁, right-top image is input₂, left-bottom is center point of synthesized movement, right-bottom is movement and hand shape.

Skeleton. This study suggests a method to make another skeleton image using the skeletal motion obtained from Kinect. For this method, existing input data requires a file with raw data and the location of each skeleton. The method of generating raw data is the same as the process in depth. The data about the skeleton position is obtained using the functions provided by the Kinect API,

and the data is generated in the same order as provided by the API so as to utilize the data. The skeleton has coordinates for each position, as shown in Fig. 7. The left side is the image of input₁, and it is the standing of a person. The right side is the image of input₂ and the person is sitting. If developers want to create a new motion by composing the following two data, developers have to set the standard of motion well and understand the need to move together. For example, if developers want to put the left hand of input₁ into the motion of input₂, applying the same coordinates will have an unnatural hand-off image. In this case, the relative movement of the hand with respect to the wrist at input₁ can be obtained and the relative position of the wrist of input₂ can be calculated to obtain the motion of the similar position with respect to the wrist. Likewise, this study tried to create a new input by synthesizing the movements of both arms of input₁ to input₂, where the relative position was the navel. Therefore, for the natural movement of the generated image arm, the relative coordinates of the shoulders, elbows, wrists, palms, thumbs, and fingers were calculated centering on the navel, and the new coordinates were calculated based on the navel of the input₂. Figure 8 shows the results of the study, showing that the lower body of input 2 is the sum of the upper arms of input₁, creating a new movement.



Fig. 7. Left is input₁ that the standing of a person, right is input₂ that sitting of a person.



Fig. 8. Synthesized skeleton image

4.4 Image Generation by Simulator

The way to create the desired movement of the user with the simulator is to use the hand shape received by using the existing Kinect. The movement of the input in a desired location and moving it to a desired location is divided into four cases, depending on the magnitude of the x and y coordinates of the starting point and the destination point. It is used to determine the type of the current movement. If the generated input moves at a given speed and arrives at the arrival point and leaves the original type, it is considered that one motion is finished and the next specified motion can be performed. Using this, continuous motion input can be created and utilized. Figure 9 shows the flow of the code-generated frames. Movements move in the order of $(0, 0)$, $(512, 424)$, $(300, 200)$, $(0, 0)$, $(200, 300)$, $(300, 200)$. Developers can code in multiple directions and consecutive movements rather than in a single direction.

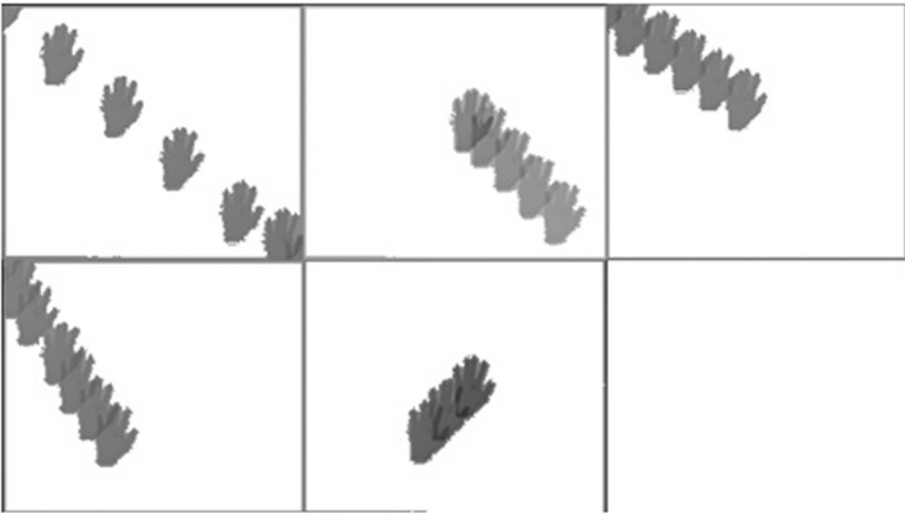


Fig. 9. Event sequence of movement by simulator

5 Conclusion

This study proposes a simulation-based development methodology that applies HCI system flow in HCI system development. Even if a limited environment such as a hardware or a tester is not available, the methodology supports developing the HCI system by using the stored event data. Therefore, a developer may develop software and hardware of the HCI system at the same time. Also, initiating the testing phase earlier, the development methodology can prevent errors to be found later and may lower the maintenance cost.

The methodology supports a developer and a tester of the HCI system, however, the motion synthesis may not cover entire requirements of the testing. Since motion events are generated based on the abstraction of hand angles and movements, a developer should consider the limitation of the motion synthesis. To ease the restrictions, a formalized modeling and simulation theory may be considered as an alternative solution.

References

1. Kim, G.J.: *Human-Computer Interaction: Fundamentals and Practice*. CRC Press Taylor & Francis Group, London (2015)
2. Nintendo Wii official homepage. <http://www.nintendo.com/wiiu/what-is-wiiu>
3. Oculus RIFT official homepage. <https://www3.oculus.com/en-us/rift/>
4. Xbox Kinect official homepage. <http://www.xbox.com/en-US/xbox-one/accessories/kinect>
5. Zhang, P., Carey, J., Te'eni, D., Tremaine, M.: Integrating human-computer interaction development into SDLC: a methodology. In: *AMCIS 2004 Proceedings* (2004)
6. Te'eni, D., Carey, J.M., Zhang, P.: *Human-Computer Interaction: Developing Effective Organizational Information System*. Wiley, New York (2006)
7. Ruparelia, N.B.: Software development lifecycle models. *ACM SIGSOFT Softw. Eng. Notes* **35**(3), 8–13 (2010)
8. Foster, I., Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco (1999)
9. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: *10th IEEE International Symposium on High Performance Distributed Computing*, pp. 181–184. IEEE Press, New York (2001)
10. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: *The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration*. Technical report, Global Grid Forum (2002)
11. National Center for Biotechnology Information. <http://www.ncbi.nlm.nih.gov>