An Architecture-Oriented Design Method for Human-Computer Interaction Systems

Yu-Chen Yang[1(✉)], Yi-Ling Lin[1], and William S. Chao[2]

[1] Department of Information Management, National Sun Yat-sen Univ., Kaohsiung, Taiwan
{ycyang,yllin}@mis.nsysu.edu.tw

[2] Association of Chinese Enterprise Architects, Taipei, Taiwan
architectchao@gmail.com

Abstract. In this paper, we propose an architecture-oriented design method for human-computer interaction systems. This design method adopts the structure-behavior coalescence (SBC) architecture as a systems model. SBC architecture design method starts from the preparation phase and then goes through the creative thinking, concept, preliminary design, and detailed design phases of SBC architecture construction. SBC architecture design method uses six fundamental diagrams to formally design the essence of a human-computer interaction system and its details at the same time. In the concept phase, architecture hierarchy diagram and framework diagram are used. In the preliminary design phase, component operation diagram and component connection diagram are used. In the detailed design phase, structure-behavior coalescence diagram and interaction flow diagram are used. With the above six diagrams, we then can effectively design the structure, behavior, and information of human-computer interaction systems; resolve uncertainties and risks caused by those non-architecture-oriented design methods.

Keywords: Architecture-Oriented design method · SBC architecture · Human-Computer interaction system

1 Introduction

In general, a human-computer interaction system is exceptionally complex that it includes multiple views such as structure, behavior, and information views [8, 10]. The systems model designs the human-computer interaction system multiple views possibly using two different methods. The first one is the non-architecture-oriented method and the second one is the architecture-oriented method [1, 6]. Non-architecture-oriented systems model respectively picks a model for each view [7, 9]. The architecture-oriented system model, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model [2, 11].

An architecture-oriented design method for human-computer interaction systems adopts the SBC architecture [3–5] as a systems model. With SBC architecture, we then can effectively design the structure, behavior, and information of human-computer interaction systems; resolve uncertainties and risks caused by those non-architecture-oriented design

F.F.-H. Nah and C.-H. Tan (Eds.): HCIB 2015, LNCS 9191, pp. 772–780, 2015.
DOI: 10.1007/978-3-319-20895-4_72

methods. Overall, SBC architecture design method helps integrate different stakeholders' works on the same track and unfold the backbone of human-computer interaction systems. The human-computer interaction system design result of SBC architecture can be used as human-computer interaction system blueprints to improve the acceptance and effectiveness of the development of human-computer interaction system.

The remaining of this paper is organized as follows. We first give a brief review of the differences between non-architecture-oriented and architecture-oriented models. Section 3 outlines the SBC architecture design method for human-computer interaction systems. Section 4 concludes the paper by summarizing the contributions of the SBC architecture design method.

2 Non-Architecture-Oriented and Architecture-Oriented Systems Models

A systems model is a virtual system, distinguished from a physical system, used to design either the physical or virtual systems. A physical system, e.g., house, tree, river, airplane, etc., exists in the physical world. A virtual system, e.g., symbol, language, diagram, software, virtual reality, thought, etc., exists in the virtual world.

A human-computer interaction system is exceptionally complex that it includes multiple views such as structure, behavior, and information views. The systems model designs the human-computer interaction system multiple views possibly using two different methods. The first one is the non-architecture-oriented method and the second one is the architecture-oriented method.

The non-architecture-oriented method respectively picks a model for each view as shown in Fig. 1, the structure view has the structure model; the behavior view has the behavior model; the information view has the information model. These multiple models are heterogeneous and unrelated of each other, thus there is no way to put them into a conformity model [7, 9].

The architecture-oriented method, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model as shown in Fig. 2. The structure, behavior, and information views are all integrated in this multiple view coalescence (MVC) systems model [1–6, 11].

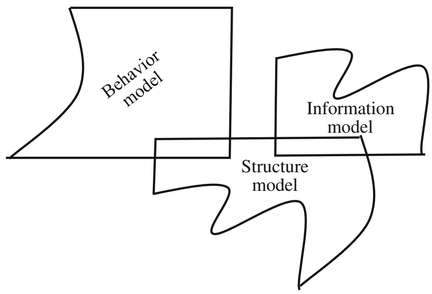

Fig. 1. The non-architecture-oriented approach

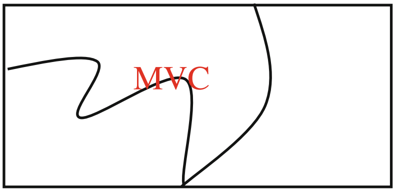

Fig. 2. The architecture-oriented approach

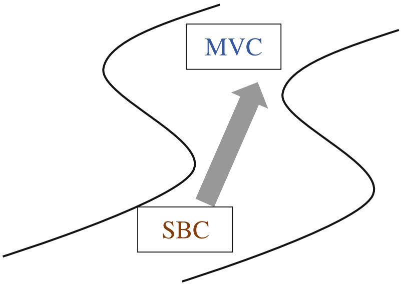

Fig. 3. SBC facilitates MVC

Figure 1 has many models. Figure 2 has only one model. Comparing Fig. 1 with Fig. 2, we unquestionably conclude that an integrated, holistic, united, coordinated, coherent, and coalescence model is more favorable than a collection of many heterogeneous and unrelated models.

Since structure and behavior views are the two most prominent ones among multiple views, integrating the structure and behavior views apparently is the best approach of integrating multiple views of a system. In other words, structure-behavior coalescence (SBC) facilitates multiple view coalescence (MVC) as shown in Fig. 3. Therefore, we claim that SBC architecture is an architecture-oriented systems model.

3 SBC Architecture Design Method for Human-Computer Interaction Systems

SBC architecture design method adopts the SBC architecture as a systems model. SBC architecture design method shall start from the preparation phase and then goes through the creative thinking, concept, preliminary design, and detailed design phases of SBC architecture construction. Each phase checks with the SBC architecture to make sure the constructed human-computer interaction system is what the users want as shown in Fig. 4.

SBC architecture design method uses six fundamental diagrams to formally design the essence of a human-computer interaction system and its details at the same time. In the concept phase, architecture hierarchy diagram (AHD) and framework diagram (FD) are used. In the preliminary design phase, component operation diagram

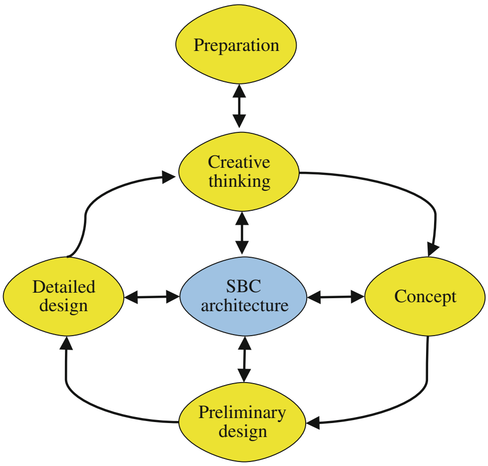

Fig. 4. SBC architecture design method

(COD) and component connection diagram (CCD) are used. In the detailed design phase, structure-behavior coalescence diagram (SBCD) and interaction flow diagram (IFD) are used.

3.1 Concept Phase

Through architecture hierarchy diagram (AHD), designers shall clearly observe the multi-level decomposition and composition of a human-computer interaction system. As an example, Fig. 5 shows that Multimedia KTV is composed of Song_Selection and Songs; Songs is composed of Song_1 and Song_2. Among them, Multimedia KTV and Songs are aggregated systems while Song_Selection, Song_1 and Song_2 are non-aggregated systems.

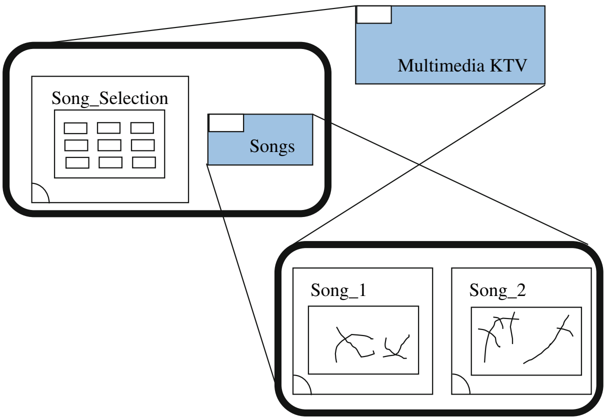

Fig. 5. AHD of the multimedia KTV

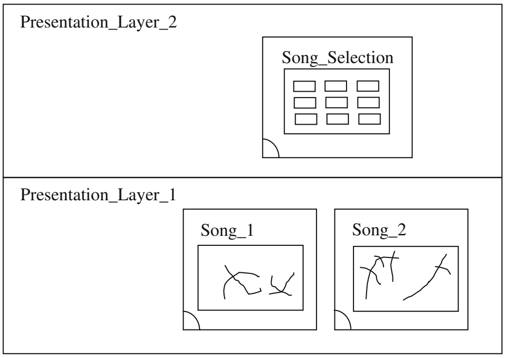

Fig. 6. FD of the multimedia KTV

Framework diagram (FD) designs the decomposition and composition of a human-computer interaction system in a multi-layer manner. Only non-aggregated systems will appear in the FD.

As an example, Fig. 6 shows a FD of the Multimedia KTV. In the figure, Presentation_Layer_2 contains the Song_Selection component; Presentation_Layer_1 contains the Song_1 and Song_2 components.

3.2 Preliminary Design Phase

For a human-computer interaction system, we use a component operation diagram (COD) to design all components' operations. Figure 7 shows a COD of the Multimedia KTV. In the figure, component Song_Selection has two operations: Select_Song_1 and Select_Song_2; component Song_1 has two operations: Broadcast_Song_1 and Sing_Song_1; component Song_Selection has two operations: Broadcast_Song_2 and Sing_Song_2.

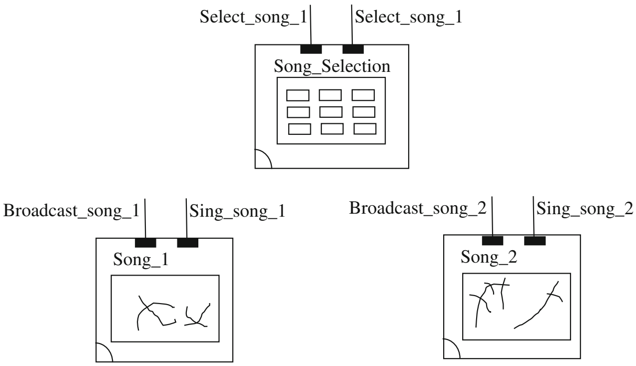

Fig. 7. COD of the multimedia KTV

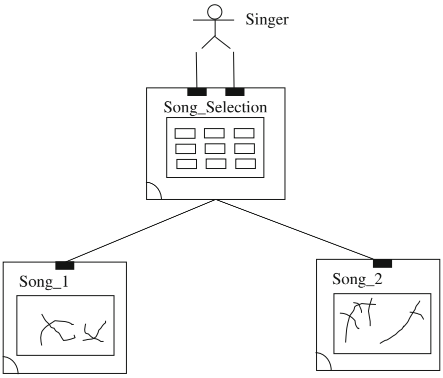

Fig. 8. CCD of the multimedia KTV

We use a component connection diagram (CCD) to design how the components and actors (in the external environment) are connected within a human-computer interaction system. Figure 8 exhibits a CCD of the Multimedia KTV.

3.3 Detailed Design Phase

In a human-computer interaction system, if the components, and among them and the external environment's actors to interact, these interactions will lead to the systems behavior. That is, "interaction" plays an important factor in coalescing structures with behaviors for a human-computer interaction system.

We use a structure-behavior coalescence diagram (SBCD) to design how the structure and behavior are integrated within a human-computer interaction system. Figure 9 exhibits a SBCD of the Multimedia KTV. In this example, an actor interacting with three components shall represent the overall systems behavior. Interactions among the Singer actor and the Song_Selection and Song_1 components generate the KalaOK_Song_1 behavior. Interactions among the Singer actor and the Song_Selection and Song_2 components generate the KalaOK_Song_2 behavior.

The overall behavior of a human-computer interaction system is the collection of all of its individual behaviors. All individual behaviors are mutually independent of each other. They tend to be executed concurrently. For example, the overall Multimedia KTV's behavior includes the KalaOK_Song_1 and KalaOK_Song_2 behaviors. In other words, the KalaOK_Song_1 and KalaOK_Song_2 behaviors are combined to produce the overall behavior of the Multimedia KTV.

The major purpose of adopting the SBC architecture design method, instead of separating the structure model from the behavior model, is to achieve one single coalesced model. In Fig. 9, designers are able to see that the systems structure and behavior coexist in a SBCD That is, in the SBCD of the Multimedia KTV, designers not only see its systems structure but also see (at the same time) its systems behavior.

The overall behavior of a human-computer interaction system consists of many individual behaviors. Each individual behavior represents an execution path. We use an

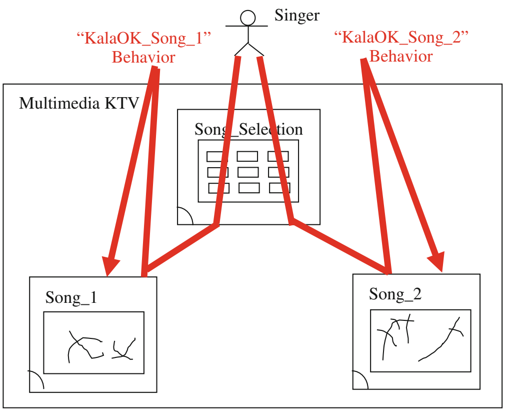

Fig. 9. SBCD of the multimedia KTV

interaction flow diagram (IFD) to design this individual behavior. The overall Multimedia KTV's behavior includes two behaviors: KalaOK_Song_1 and KalaOK_Song_2.

Figure 10 shows the IFD of the KalaOK_Song_1 behavior. First, actor Singer interacts with the Song_Selection component through the Select_Song_1 operation call interaction. Next, component Song_Selection interacts with the Song_1 component through the Broadcast_Song_1 operation call interaction. Finally, actor Singer interacts with the Song_1 component through the Sing_Song_1 operation call interaction.

Figure 11 shows the IFD of the KalaOK_Song_2 behavior. First, actor Singer interacts with the Song_Selection component through the Select_Song_2 operation call interaction. Next, component Song_Selection interacts with the Song_2 component through the Broadcast_Song_2 operation call interaction. Finally, actor Singer interacts with the Song_2 component through the Sing_Song_2 operation call interaction.

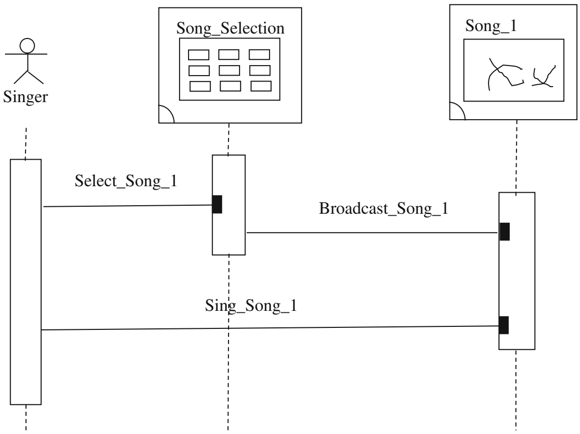

Fig. 10. IFD of the "KalaOK_Song_1" behavior

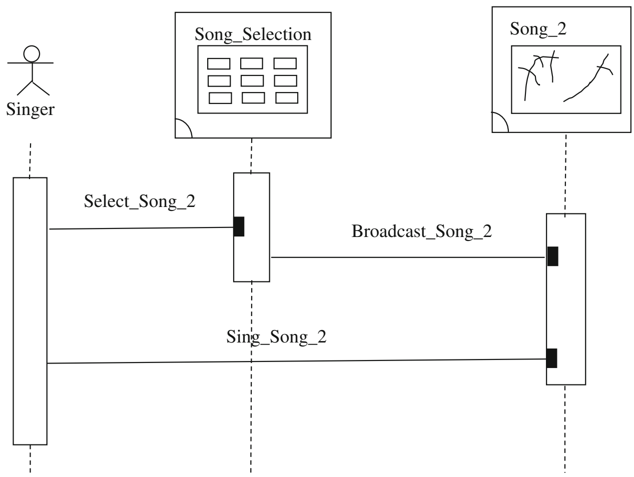

Fig. 11. IFD of the "KalaOK_Song_2" behavior

4 Conclusions

A human-computer interaction system is very complex that it includes multiple views such as structure, behavior, and information views. The systems model designs the human-computer interaction system multiple views possibly using two different methods. The first one is the non-architecture-oriented method and the second one is the architecture-oriented method.

Non-architecture-oriented systems model respectively picks a model for each view. These multiple models are heterogeneous and unrelated of each other, thus there is no way to put them into a conformity model. Architecture-oriented systems model, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model. The structure, behavior, and information views are all integrated in this multiple view coalescence (MVC) systems model.

Since structure and behavior views are the two most prominent ones among multiple views, integrating the structure and behavior views apparently is the best approach of integrating those multiple views of a system. In other words, structure-behavior coalescence (SBC) facilitates multiple view coalescence (MVC). Therefore, we claim that SBC architecture is an architecture-oriented systems model.

SBC architecture design method adopts the SBC architecture as a systems model. SBC architecture design method starts from the preparation phase and then goes through the creative thinking, concept, preliminary design, and detailed design phases of SBC architecture construction. SBC architecture design method uses six fundamental diagrams to formally design the essence of a human-computer interaction system and its details at the same time. In the concept phase, architecture hierarchy diagram and framework diagram are used. In the preliminary design phase, component operation diagram and component connection diagram are used. In the detailed design phase, structure-behavior coalescence diagram and interaction flow diagram are used.

With these six diagrams, we then can effectively design the structure, behavior, and information of human-computer interaction systems; resolve uncertainties and risks caused by those traditional non-architecture-oriented design methods. Overall, SBC architecture design method helps integrate different stakeholders' works on the same track and unfold the backbone of human-computer interaction systems. The human-computer interaction system design result of SBC architecture can be used as human-computer interaction system blueprints to improve the acceptance and effectiveness of the development of human-computer interaction system.

References

1. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice, 2nd edn. Addison-Wesley, Boston (2003)
2. Bernard, S.: An Introduction to Enterprise Architecture, 2nd edn. AuthorHouse, Bloomington (2005)
3. Chao, W.S.: General Systems Theory 2.0: General Architectural Theory Using the SBC Architecture. CreateSpace Independent Publishing (2014)
4. Chao, W.S.: Systems Thinking 2.0: Architectural Thinking Using the SBC Architecture Description Language. CreateSpace Independent Publishing (2014)
5. Chao W.S.: Systems Modeling and Architecting: Structure-Behavior Coalescence for Systems Architecture. CreateSpace Independent Publishing 2014
6. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., Stafford, J.: Documenting Software Architectures: Views and Beyond, 2nd edn. Addison Wesley, Boston (2010)
7. Dennis, A., Wixom, B.H., Roth, R.M.: Systems Analysis and Design, 4th edn. John Wiley & Sons, Hoboken (2008)
8. Jacko, J.A.: Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. CRC Press, Boca Raton (2012)
9. Kendall, K.E., Kendall, J.E.: Systems Analysis and Design, 8th edn. Prentice Hall, Upper Saddle River (2010)
10. MacKenzie, I.S.: Human-Computer Interaction: An Empirical Research Perspective. Morgan Kaufmann Publishers, Burlington (2013)
11. Maier, M.W.: The Art of Systems Architecting, 3rd edn. CRC Press, Boca Raton (2009)