PST20130905130708VED

An Architecture-Oriented Design Method For Innovative Service Systems

Shuh-Ping Sun* William S. Chao**

*Professor, Department of Digital Media Design, I-Shou University

(No. 1, Sec. 1, Syuecheng Rd., Dashu District, Kaohsiung City 84001, Taiwan, R.O.C.

07-6577711 Ext.8451、spsun@isu.edu.tw)

**President, Association of Chinese Enterprise Architects

Abstract

In this paper, we propose an architecture-oriented design method for innovative service 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 planning, preliminary design, and detailed design phases of SBC architecture construction. In the planning 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. Overall, SBC architecture design method helps integrate different stakeholders' works on the same track and unfold the backbone of an innovative service system; resolve uncertainties and risks caused by those non-architecture-oriented design methods.

1. Introduction

In general, an innovative service system is exceptionally complex that it includes multiple views such as structure, behavior, and information views. The systems model designs the innovative service 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, 8]. Architecture-oriented systems model, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model[2, 9].

An architecture-oriented design method for innovative service systems adopts the SBC architecture[3, 4, 5] as a systems model. With SBC architecture, we then can effectively design the structure, behavior, and information of innovative service systems; resolve uncertainties and risks caused by those non-architecture-oriented design methods. Overall, SBC architecture design method helps integrate different stakeholders' works on the same track and unfold the backbone of innovative service systems. The innovative service system design result of SBC architecture can be used as innovative service system schemes to improve the acceptance and effectiveness of the development of innovative service system.

2. Non-architecture-oriented and Architecture-oriented Systems Models

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

Figure 1 shows a physical system in which there are two buildings located in the upper left side and right underneath. The upper left building is Seattle Hotel and the right underneath building is Dallas Theater.

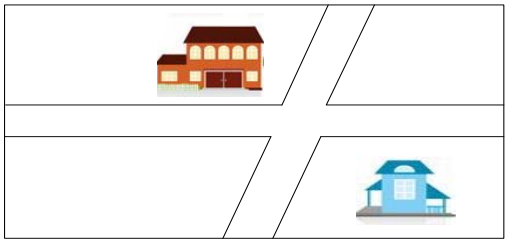

Figure 1 A Physical System

To design the physical system in Figure 1, we

may then obtain a map as shown in Figure 2. The map is a kind of systems model used to design the physical system.

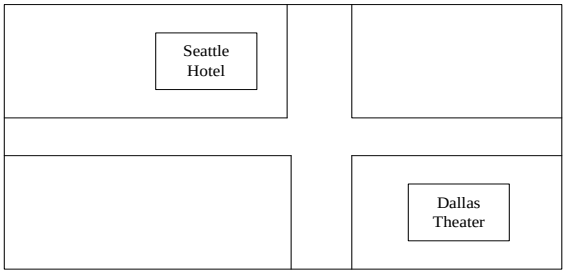

Figure 2 Map as a Systems Model

Besides designing systems in the physical world, a systems model can also design systems in the conceptual world. The conceptual world includes a software system, a virtual reality, or a thought within a person's mind, etc. Figure 3 demonstrates that a manager is planning a sale task. Planning a sale task, being a thought inside a person's mind, belongs to the conceptual world.

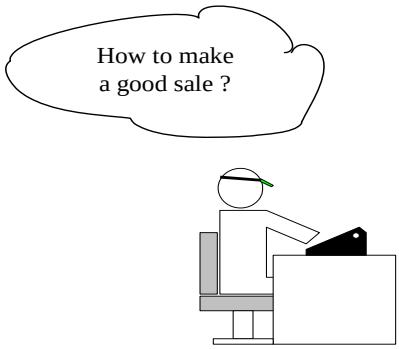

Figure 3 Thought inside a Person's Mind

To design the thought within a person's mind in Figure 3, we may then use a sale chart as shown in Figure 4. The sale chart is a kind of systems model used to design a person's thought.

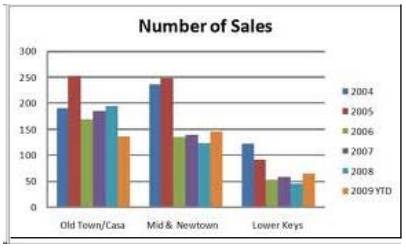

Figure 4 Sale Chart as a System Model

An innovative service system is exceptionally complex that it includes multiple views such as structure, behavior, and information views. The systems model designs the innovative service 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 Figure 5, 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, 8].

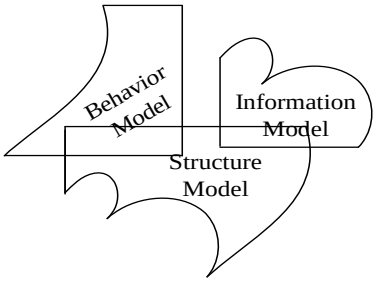

Figure 5 The Non-architecture-oriented Approach

The architecture-oriented method, instead of picking many heterogeneous and unrelated models, will use only one single coalescence model as shown in Figure 6. The structure, behavior, and information views are all integrated in this multiple view coalescence (MVC) systems model[1, 2, 3, 4, 5, 6, 9, 10].

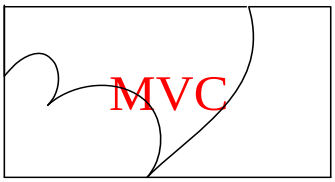

Figure 6 The Architecture-oriented Approach

Figure 5 has many models. Figure 6 has only one model. Comparing Figure 5 with Figure 6, 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 Figure 7. Therefore, we claim that SBC architecture is an architecture-oriented systems model.

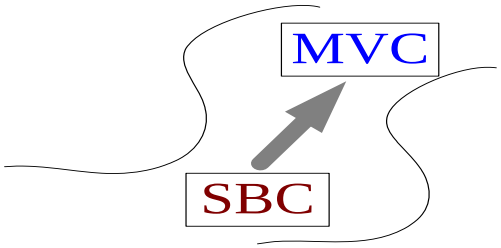

Figure 7 SBC Facilitates MVC

3. SBC Architecture Design Method for Innovative service 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, planning, analysis, and design phases of SBC architecture construction. Each phase checks with the SBC architecture to make sure the constructed innovative service system is what the users want as shown in Figure 8.

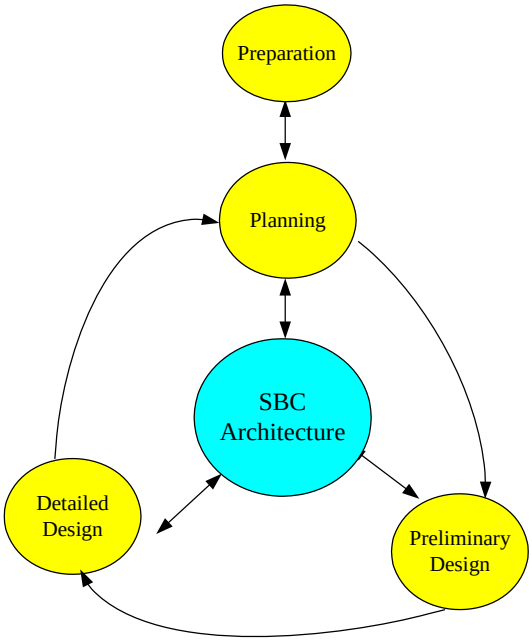

Figure 8 SBC Architecture Design Method For Innovative Service Systems

SBC architecture design method uses six fundamental diagrams to formally design the essence of an innovative service system and its details at the same time. In the planning phase, architecture hierarchy diagram (AHD) and framework diagram (FD) are used. In the analysis phase, component operation diagram (COD) and component connection diagram (CCD) are used. In the design phase, structure-behavior coalescence diagram (SBCD) and interaction flow diagram (IFD) are used.

3.1. Planning Phase

Through the architecture hierarchy diagram, designers shall clearly observe the multi-level decomposition and composition of an innovative service system. As an example, Figure 9 shows that *Health Care Service* is composed of *BloodPressure_GUI*, *HealthReport_GUI*, *HealthAnalysis_GUI*, and *Data_Layer+Technology_Layer*; *Data_Layer+Technology_Layer* is composed of *Health Care Service Database* and *Technology_Layer*; *Technology_Layer* is composed of *Health Care Service Hardware*. Among them, *Health Care Service*, *Data_Layer+Technology_Layer*, and *Technology_Layer* are aggregated systems while *BloodPressure_GUI*, *HealthReport_GUI*, *HealthAnalysis_GUI*, *Health Care Service Database*, and *Health Care Service Hardware* are non-aggregated systems.

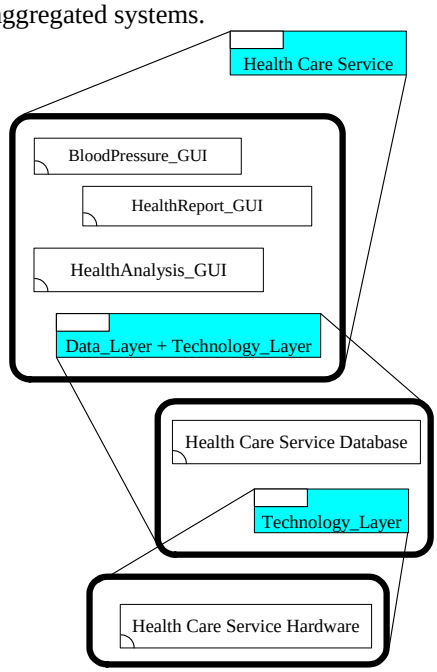

Figure 9 AHD of *Health Care Service*

The framework diagram (FD) designs the decomposition and composition of an innovative service system in a multi-layer manner. Only non-aggregated systems will appear in the FD.

As an example, Figure 10 shows the FD of *Health Care Service*. In the figure, *Application_Layer* contains the components *BloodPressure_GUI*, *HealthReport_GUI*, and *HealthAnalysis_GUI*; *Data_Layer* contains the component *Health Care Service Database*; *Technology_Layer* contains the component *Health Care Service Hardware*.

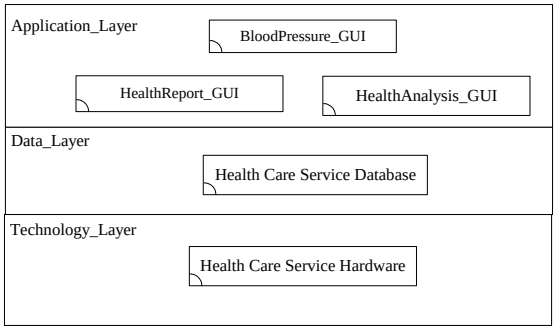

Figure 10 FD of *Health Care Service*

3.2. Preliminary Design Phase

For an innovative service system, we use component operation diagram (COD) to design all components' operations. Figure 11 shows the COD of *Health Care Service*. In the figure, component *BloodPressure_GUI* has one operation: *Measurement_Input*; component *HealthReport_GUI* has one operation: *Status_Output*; component *HealthAnalysis_GUI* has one operation: *Analysis_Output*; component *Health Care Service Database* has one operation: *Health Care Service SQL_Query*; component *Health Care Service Hardware* has one operation: *Blood_Measure*.

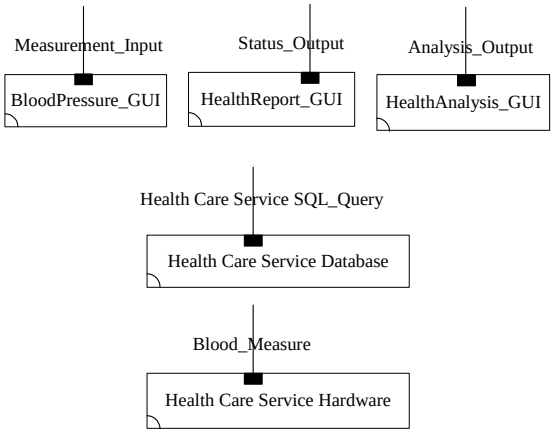

Figure 11 COD of *Health Care Service*

We use the component connection diagram (CCD) to design how the components and actors (in the external environment) are connected within an innovative service system. Figure 12 exhibits the CCD of *Health Care Service*.

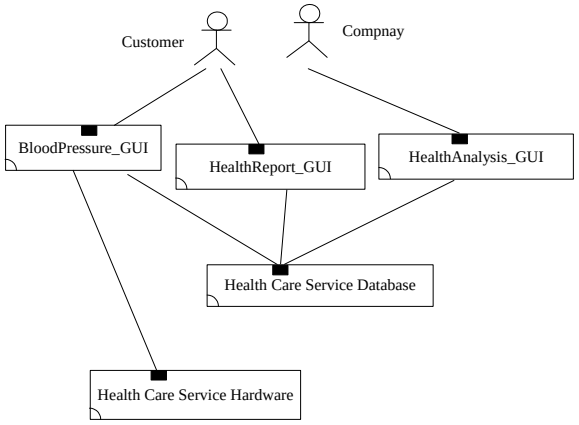

Figure 12 CCD of *Health Care Service*

3.3. Detailed Design Phase

In an innovative service 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 an innovative service system.

We use the structure-behavior coalescence diagram (SBCD) to design how the structure and behavior are integrated within an innovative service system. Figure 13 exhibits the SBCD of *Health Care Service*. In this example, an actor interacting with five components shall represent the overall systems behavior. Interactions among the actor *Customer* and the components *BloodPressure_GUI*, *Health Care Service Database*, and *Health Care Service Hardware* generate the behavior *BloodPressure_Measure*. Interactions among the actor *Customer* and the components *HealthReport_GUI* and *Health Care Service Database* generate the behavior *HealthStatus*. Interactions among the actor *Company* and the components *HealthAnalysis_GUI* and *Health Care Service Hardware* generate the behavior *HealthAnalysis*.

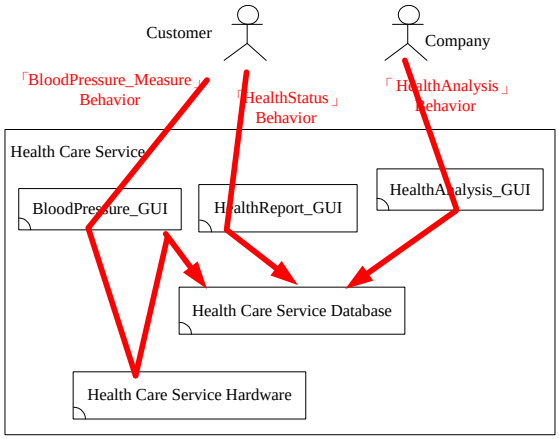

Figure 13 SBCD of *Health Care Service*

The overall behavior of an innovative service 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 *Health Care Service's* behavior includes behaviors *BloodPressure_Measure*, *HealthStatus*, and *HealthAnalysis*. In other words, the behaviors *BloodPressure_Measure*, *HealthStatus*, and *HealthAnalysis* are combined to produce the overall behavior of *Health Care Service*.

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 Figure 13, designers are able to see that the systems structure and behavior coexist in the SBCD. That is, in the SBCD of *Health Care Service*, designers not only see its systems structure but also see (at the same time) its systems behavior.

The overall behavior of an innovative service system consists of many individual behaviors. Each individual behavior represents an execution path. We use interaction flow diagram (IFD) to design this individual behavior. The overall *Health Care Service's* behavior includes three behaviors: *BloodPressure_Measure*, *HealthStatus*, and *HealthAnalysis*.

Figure 14 shows the IFD of the behavior *BloodPressure_Measure*. First, actor *Customer* interacts with the component *BloodPressure_GUI* through the operation call interaction *Measurement_Input*. Next, component *BloodPressure_GUI* interacts with the component *Health Care Service Hardware* through the operation call interaction *BloodPressure_Measure*. Finally, component *BloodPressure_GUI* interacts with the component *Health Care Service Database* through the operation call interaction *Health Care Service SQL_Query*.

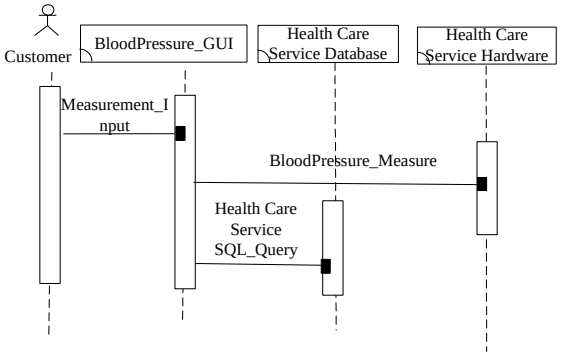

Figure 14 IFD of the 「Blood_Measure」 Behavior

Figure 15 shows the IFD of the behavior *HealthStatus*. First, actor *Customer* interacts with the component *HealthReport_GUI* through the operation call interaction *Status_Output*. Finally, component *HealthReport_GUI* interacts with the component *Health Care Service Database* through the operation call interaction *Health Care Service SQL_Query*.

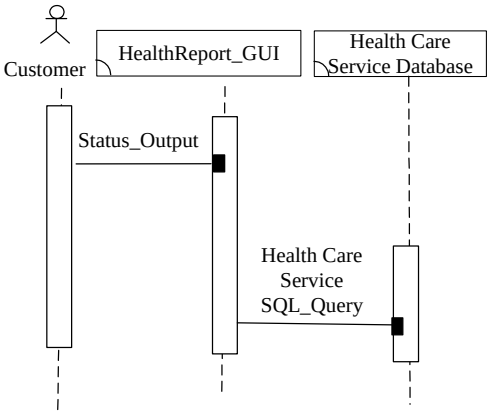

Figure 15 IFD of the 「HealthStatus」 Behavior

Figure 16 shows the IFD of the behavior *HealthAnalysis*. First, actor *Company* interacts with the component *HealthAnalysis_GUI* through the operation call interaction *Analysis_Output*. Finally, component *HealthAnalysis_GUI* interacts with the component *Health Care Service Database* through the operation call interaction *Health Care Service SQL_Query*.

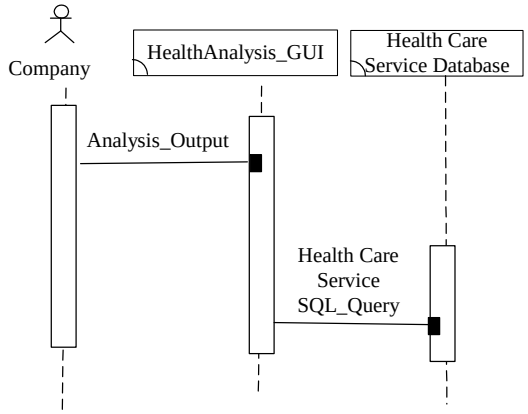

Figure 16 IFD of the「HealthAnalysis」Behavior

4. Conclusions

An innovative service system is very complex that it includes multiple views such as structure, behavior, and information views. The systems model designs the innovative service 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 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 for innovative service systems adopts the SBC architecture as a systems model. SBC architecture design method starts from the preparation phase and then goes through the creative thinking, planning, preliminary design, and detailed design phases of SBC architecture construction. SBC architecture design method uses six fundamental diagrams to formally design the essence of an innovative service system and its details at the same time. In the planning 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 innovative service 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 innovative service systems. The innovative service system design result of SBC architecture can be used as innovative service system schemes to improve the acceptance and effectiveness of the development of innovative service system.

5. References

1. Bass, L. et al., Software Architecture in Practice, 2nd Edition, Addison-Wesley, 2003.

2. Bernard, S., An Introduction To Enterprise Architecture, 2nd Edition, AuthorHouse, 2005.

3. Chao, W. S. and J. Moore and C. S. Chang, System Analysis and Design: SBC Software Architecture in Practice, LAP Lambert Academic Publishing, 2009.

4. Chao, W. S., Software Architecture: SBC Architecture at Work, National Sun Yat-sen University Press, 2011.

5. Chao, W. S. and S. P. Sun and M. C. Harn, Systems Architecture: Hardware, Software, Enterprise, Knowledge, Thinking, I-Shou University System Architecture Research Center Publishing, 2013.

6. Clements, P. et al., Documenting Software Architectures: Views and Beyond, 2nd Edition, Addison Wesley, 2010.

7. Dennis, A. et al., Systems Analysis and Design, 4th Edition, Wiley, 2008.

8. Kendall, K. et al., Systems Analysis and Design, 8th Edition, Prentice Hall, 2010.

9. Maier, M. W., The Art of Systems Architecting, 3rd Edition, CRC Press, 2009.

10. Shuh-Ping Sun et al., "An Architecture-Oriented Design Method For Interactive Multimedia Systems", The 5th International Congress of International Association of Societies of Design Research (IASDR), Tokyo, Japan, 2013.